

Docket No.: **488-028**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
PATENT OPERATIONS

In re Application of:

**Akira Hasuike**

Serial No.: N/A

Filed: N/A

For: **"IMAGE DISPLAY METHOD"**

)  
)  
) Group Art Unit:  
)  
) Examiner: N/A  
)  
)

New York, NY 10036

April 14, 2004

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**CLAIM FOR CONVENTION PRIORITY**

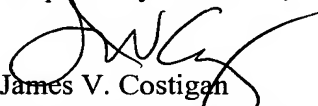
SIR:

In the matter of the above-identified application and under the provisions of 35 U.S.C. §119 Applicants claim the benefit of the following prior applications:

Application filed in: **Japan**  
Application No.: **2003-112998**  
Filed: **April 17, 2003**

Pursuant to the Claim to Priority, Applicants submit a duly Certified Copy of said foreign application.

Respectfully submitted,

  
James V. Costigan  
Registration No. 25,669

**MAILING ADDRESS:**  
**HEDMAN & COSTIGAN, P.C.**  
1185 Avenue of the Americas  
New York, NY 10036  
(212) 302-8989

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

"Express Mail" Mailing Label # EL988370626US

Date of Deposit April 14, 2004

I hereby certify that this paper or fee is being deposited with the United States Postal Service by "Express Mail Post Office To Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to

Commissioner of Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

on April 14, 2004

  
James V. Costigan, Registration No. 25,669

(Typed or printed name of person mailing)



日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application: 2 0 0 3 年 4 月 1 7 日  
April 17, 2003

出 願 番 号  
Application Number: 特 願 2 0 0 3 - 1 1 2 9 9 8  
[ST. 10/C]: [ J P 2 0 0 3 - 1 1 2 9 9 8 ]

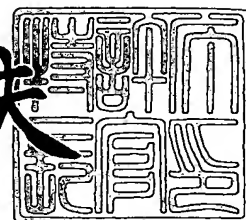
出 願 人  
Applicant(s): 株式会社サピエンス  
Sapience Corporation



2 0 0 4 年 4 月 1 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 4 - 3 0 2 7 1 6 8

【書類名】 特許願

【整理番号】 SAP0208

【提出日】 平成15年 4月17日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 13/00

【発明者】

    【住所又は居所】 東京都豊島区南大塚 3 - 2 0 - 6 株式会社サピエンス  
                                内

    【氏名】 蓮池 曜

【特許出願人】

    【識別番号】 594164379

    【氏名又は名称】 株式会社 サピエンス

【代理人】

    【識別番号】 100090228

    【弁理士】

    【氏名又は名称】 加藤 邦彦

    【電話番号】 03(3359)9553

【手数料の表示】

    【予納台帳番号】 062422

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 画像表示方法

【特許請求の範囲】

【請求項 1】

ビューアの表示領域よりも大きい画像をサーバからダウンロードしてビューアに表示する方法であって、サーバから該画像を複数の領域に分割した画像ごとに送信可能にし、該画像とビューアの表示領域との相対位置に応じてビューアの表示領域に少なくとも一部が入る分割画像を判断し、サーバから該当する分割画像を優先的に送信し、ビューアに表示する画像表示方法。

【請求項 2】

ビューアの表示領域よりも大きい画像をサーバからダウンロードしてビューアに表示する方法であって、該画像を複数の領域に分割してサーバに予め用意し、該画像とビューアの表示領域との相対位置に応じてビューアの表示領域に少なくとも一部が入る分割画像を判断し、サーバから該当する分割画像を優先的に送信し、ビューアに表示する画像表示方法。

【請求項 3】

ビューアの表示領域よりも大きい画像をサーバからダウンロードしてビューアに表示する方法であって、該画像を、ビューアの表示領域よりも横方向もしくは縦方向の一方向または横方向および縦方向の両方向の長さがそれぞれ短い複数の領域に分割してサーバに予め用意し、該画像とビューアの表示領域との相対位置に応じてビューアの表示領域に少なくとも一部が入る複数の分割画像を判断し、サーバから該当する複数の分割画像を優先的に送信し、ビューアに元の並び状態に配列して表示する画像表示方法。

【請求項 4】

ビューアの表示領域よりも大きい画像をサーバからダウンロードしてビューアに表示する方法であって、該画像を複数の領域に分割してサーバに予め用意し、ビューアは該画像とビューアの表示領域との相対位置に応じてビューアの表示領域に少なくとも一部が入る分割画像を判断して該分割画像をサーバに優先的に要求し、サーバは該要求に応じた分割画像を優先的に送信し、ビューアは受け取っ

た分割画像を表示する画像表示方法。

**【請求項 5】**

ビューアの表示領域よりも大きい画像をサーバからダウンロードしてビューアに表示する方法であって、該画像を、ビューアの表示領域よりも横方向もしくは縦方向の一方向または横方向および縦方向の両方向の長さがそれぞれ短い複数の領域に分割してサーバに予め用意し、ビューアは該画像とビューアの表示領域との相対位置に応じてビューアの表示領域に少なくとも一部が入る複数の分割画像を判断して該分割画像をサーバに優先的に要求し、サーバは該要求に応じた複数の分割画像を優先的に送信し、ビューアは受け取った分割画像を元の並び状態に配列して表示する画像表示方法。

**【請求項 6】**

ビューアの表示領域に少なくとも一部が入る分割画像のほか、該分割画像の領域の周囲に隣接する分割画像を含む、ビューアの表示領域に対し所定の位置関係にある限定された範囲の画像領域に含まれる分割画像を併せて判断し、サーバから該当する分割画像を併せて優先的に送信する請求項 1 から 5 のいずれかに記載の画像表示方法。

**【請求項 7】**

前記分割画像がビューアに既にダウンロードされて保存されている分割画像であるかどうかを判断し、既に保存されている分割画像である場合は、サーバから再度ダウンロードすることなく、該保存されている分割画像を読み出して表示する請求項 1 から 6 のいずれかに記載の画像表示方法。

**【請求項 8】**

前記分割画像が、前記画像を横方向もしくは縦方向の一方向にまたは横方向および縦方向の両方向に格子状に分割した画像である請求項 1 から 7 のいずれかに記載の画像表示方法。

**【請求項 9】**

前記格子が、前記画像の左端位置を始点として所定ピクセル数ごとに横方向に分割して形成されたもの、もしくは、該画像の上端位置を始点として所定ピクセル数ごとに縦方向に分割して形成されたもの、または、該画像の左端位置を始点

として所定ピクセル数ごとに横方向に分割しかつ該画像の上端位置を始点として所定ピクセル数ごとに縦方向に分割して形成されたものである請求項 8 記載の画像表示方法。

【請求項 1 0】

ビューアの表示領域に少なくとも一部が入る分割画像を含む、ビューアの表示領域に対し所定の位置関係にある限定された範囲の画像領域に含まれる分割画像を、個々に当てはめて表示する表示領域に相当する所定数の枠要素をビューアに設定し、該各枠要素に、該当する位置の分割画像をそれぞれ当てはめて表示し、かつ、画像の相対移動に伴いビューアの表示領域から離れる分割画像を判断して枠要素への当てはめを解除し、ビューアの表示領域に近づく分割画像を判断して枠要素に新たに当てはめる請求項 1 から 9 のいずれかに記載の画像表示方法。

【請求項 1 1】

前記各枠要素に固有の識別情報が付与され、前記画像の相対移動に伴いビューアの表示領域から離れる分割画像の当てはめを解除する枠要素に、ビューアの表示領域に近づく分割画像を新たに当てはめる請求項 1 0 記載の画像表示方法。

【請求項 1 2】

前記各枠要素に固有の識別情報が付与され、ビューアは該固有の識別情報に、ビューアの表示領域における該枠要素の表示位置の情報と、該枠要素に当てはめる分割画像の識別情報を関連づけて保持し、ビューアは該情報に基づき、各枠要素に当てはめる分割画像を、ビューアの表示領域の所定位置に表示する請求項 1 0 または 1 1 記載の画像表示方法。

【請求項 1 3】

前記分割画像の識別情報が、画像全体におけるその番地に相当する情報をもって該画像全体における識別情報としてなり、ビューアは該分割画像の識別情報をもってサーバに該分割画像を要求し、サーバは該識別情報に該当する分割画像を判別してビューアに送信する請求項 1 2 記載の画像表示方法。

【請求項 1 4】

前記分割画像の識別情報が、各分割画像のファイル形式を識別する情報を有しない請求項 1 3 記載の画像表示方法。

**【請求項 15】**

前記所定数の枠要素全体でブロックを構成し、画像の相対移動が指示された時に、ビューアは、該ビューアの表示領域の原点に対する該ブロックの原点の移動すべき座標を演算し、該演算により求められたブロックの原点の座標に基づき、前記各枠要素の原点の移動すべき座標をそれぞれ演算し、該演算により求められた座標に各枠要素の原点をそれぞれ移動することにより、前記画像の相対移動を実現する請求項 10 から 14 のいずれかに記載の画像表示方法。

**【請求項 16】**

前記分割画像が前記画像を横方向に格子状に分割した画像でありかつ前記枠要素が横方向に所定数の連続した分割画像を当てはめるものであり、もしくは、前記分割画像が前記画像を縦方向に格子状に分割した画像でありかつ前記枠要素が縦方向に所定数の連続した分割画像を当てはめるものであり、または、前記分割画像が前記画像を横方向および縦方向に格子状に分割した画像でありかつ前記枠要素が横方向に所定数の連続した分割画像もしくは縦方向に所定数の連続した分割画像または横方向に所定数で縦方向に所定数の連続した分割画像を当てはめるものである請求項 10 から 15 のいずれかに記載の画像表示方法。

**【請求項 17】**

同じ内容の画像を複数の倍率でかつ倍率ごとにそれぞれ複数の領域に分割してサーバから送信可能にし、ビューアから指示される倍率の画像について請求項 1 から 16 のいずれかに記載の方法を実行する画像表示方法。

**【請求項 18】**

倍率にかかわらず、画像を横方向もしくは縦方向または横方向および縦方向に分割するピクセル数が同じである請求項 17 記載の画像表示方法。

**【請求項 19】**

1つの倍率で画像を表示している状態で該画像上の任意の位置にポインタを当てて倍率変更操作をした場合に、該ポインタが位置する画像上の位置を不動点として変更後の倍率の画像を表示する請求項 17 または 18 記載の画像表示方法。

**【請求項 20】**

同じ内容の画像を複数の倍率でかつ倍率ごとにそれぞれ複数の領域に分割して

サーバから送信可能にし、ビューアから指示される倍率の画像について請求項 10 から 16 のいずれかに記載の方法を実行する画像表示方法であって、倍率にかかわらず、前記枠要素の数を同じにしてなる画像表示方法。

**【請求項 21】**

前記ビューアが Web ブラウザであり、該 Web ブラウザでの各演算がサーバから送信される HTML に格納されている JavaScript（登録商標）に基づき実行される請求項 1 から 20 のいずれかに記載の画像表示方法。

**【請求項 22】**

前記枠要素が、サーバから送信される HTML に記述されている <DIV> タグを使用して設定されている請求項 21 記載の画像表示方法。

**【請求項 23】**

前記サーバから送信される HTML に、前記分割画像ごとのファイル名及び／またはファイル形式等の属性の情報が組み込まれていない請求項 21 または 22 記載の画像表示方法。

**【請求項 24】**

前記画像が、異なるファイル形式の分割画像が混在して構成されている請求項 1 ～ 23 のいずれかに記載の画像表示方法。

**【発明の詳細な説明】**

**【0001】**

**【発明の属する技術分野】**

この発明は、ビューアの表示領域よりも大きい画像をサーバからダウンロードしてビューアに表示する方法に関する。

**【0002】**

**【従来の技術】**

インターネット技術の発展により、世界中からあらゆる情報をブラウザで閲覧することができるようになった。画像情報も同様にブラウザで閲覧できる。これは、Web サーバ上に、JPEG ファイル等の画像ファイルを表示することを記述した HTML ファイルを保存しておくことにより実現される。インターネット技術を利用した新聞紙面画像の配信および閲覧も実用化されている。インターネ



ット技術を利用した新聞紙面画像の配信・閲覧技術としては、例えば下記特許文献 1 に記載のものがある。

【0 0 0 3】

【特許文献 1】

特開 2 0 0 2 - 2 3 6 7 0 1 号公報

【0 0 0 4】

【発明が解決しようとする課題】

インターネット技術を利用して新聞紙面等の横縦のサイズ（画素数）の大きい画像をダウンロードしてブラウザに表示するのに、従来は長い待ち時間を要していた。

【0 0 0 5】

また、インターネット技術を利用した新聞紙面画像の配信・閲覧は、従来は、ブラウザから起動される専用のソフトウェア（プラグインソフトウェア、専用ビューア）を別途入手して実現していた。よく知られているVeriSign（登録商標）などの認証機関によって確認されるソフトウェアさえも、製作した会社を証明するものであって、ソフトウェア自体を保証するものではないことからソフトウェアの使用はユーザの責任になる。実際、ブラウザには新しい技術が続々と追加されるので、一般ユーザがブラウザの動きを理解し、自己責任でソフトウェアを選別して使用することは非常に難しい。このことから、一般ユーザはプラグインソフトウェアの使用に不安を感じる事が多く、セキュリティやウィルス、ハードディスクの破損などの問題からプラグインソフトウェアの使用を拒否するユーザも少なくない。

【0 0 0 6】

この発明は、上述の点に鑑みてなされたもので、横縦のサイズ（画素数）が大きい画像をダウンロードして、少ない待ち時間でビューアに表示できるようにした画像表示方法を提供しようとするものである。また、この発明は、併せて、プラグインソフトウェアなしで、該画像表示方法を実現する方法を提供しようとするものである。

【0 0 0 7】

**【課題を解決するための手段】**

この発明の画像表示方法は、ビューアの表示領域よりも大きい画像をサーバからダウンロードしてビューアに表示する方法であって、サーバから該画像を複数の領域に分割した画像ごとに送信可能にし、該画像とビューアの表示領域との相対位置に応じてビューアの表示領域に少なくとも一部が入る分割画像を判断し、サーバから該当する分割画像を優先的に送信し、ビューアに表示するものである。

**【0008】**

この発明の画像表示方法は、ビューアの表示領域よりも大きい画像をサーバからダウンロードしてビューアに表示する方法であって、該画像を複数の領域に分割してサーバに予め用意し、該画像とビューアの表示領域との相対位置に応じてビューアの表示領域に少なくとも一部が入る分割画像を判断し、サーバから該当する分割画像を優先的に送信し、ビューアに表示するものである。

**【0009】**

この発明の画像表示方法は、ビューアの表示領域よりも大きい画像をサーバからダウンロードしてビューアに表示する方法であって、該画像を、ビューアの表示領域よりも横方向もしくは縦方向の一方向または横方向および縦方向の両方向の長さがそれぞれ短い複数の領域に分割してサーバに予め用意し、該画像とビューアの表示領域との相対位置に応じてビューアの表示領域に少なくとも一部が入る複数の分割画像を判断し、サーバから該当する複数の分割画像を優先的に送信し、ビューアに元の並び状態に配列して表示するものである。

**【0010】**

この発明の画像表示方法は、ビューアの表示領域よりも大きい画像をサーバからダウンロードしてビューアに表示する方法であって、該画像を複数の領域に分割してサーバに予め用意し、ビューアは該画像とビューアの表示領域との相対位置に応じてビューアの表示領域に少なくとも一部が入る分割画像を判断して該分割画像をサーバに優先的に要求し、サーバは該要求に応じた分割画像を優先的に送信し、ビューアは受け取った分割画像を表示するものである。

**【0011】**

この発明の画像表示方法は、ビューアの表示領域よりも大きい画像をサーバからダウンロードしてビューアに表示する方法であって、該画像を、ビューアの表示領域よりも横方向もしくは縦方向の一方向または横方向および縦方向の両方向の長さがそれぞれ短い複数の領域に分割してサーバに予め用意し、ビューアは該画像とビューアの表示領域との相対位置に応じてビューアの表示領域に少なくとも一部が入る複数の分割画像を判断して該分割画像をサーバに優先的に要求し、サーバは該要求に応じた複数の分割画像を優先的に送信し、ビューアは受け取った分割画像を元の並び状態に配列して表示するものである。

#### 【0012】

この発明の画像表示方法によれば、画像とビューアの表示領域との相対位置に応じてビューアの表示領域に少なくとも一部が入る分割画像を判断し、サーバから該当する分割画像を優先的に送信するようにしたので、画像全体を送信して表示する場合に比べて、少ない待ち時間でビューアに表示することができる。

#### 【0013】

この発明では、ビューアの表示領域に少なくとも一部が入る分割画像のほか、該分割画像の領域の周囲に隣接する分割画像を含む、ビューアの表示領域に対し所定の位置関係にある限定された範囲の画像領域に含まれる分割画像を併せて判断し、サーバから該当する分割画像を併せて優先的に送信することができる。

#### 【0014】

また、前記分割画像がビューアに既にダウンロードされて保存されている分割画像であるかどうかを判断し、既に保存されている分割画像である場合は、サーバから再度ダウンロードすることなく、該保存されている分割画像を読み出して表示することができる。

#### 【0015】

また、前記分割画像は、前記画像を横方向もしくは縦方向の一方向にまたは横方向および縦方向の両方向に格子状に分割した画像とすることができる。この場合、前記格子は、前記画像の左端位置を始点として所定ピクセル数ごとに横方向に分割して形成されたもの、もしくは、該画像の上端位置を始点として所定ピクセル数ごとに縦方向に分割して形成されたもの、または、該画像の左端位置を始

点として所定ピクセル数ごとに横方向に分割しかつ該画像の上端位置を始点として所定ピクセル数ごとに縦方向に分割して形成されたものとすることができる。

【0016】

また、ビューアの表示領域に少なくとも一部が入る分割画像を含む、ビューアの表示領域に対し所定の位置関係にある限定された範囲の画像領域に含まれる分割画像を、個々に当てはめて表示する表示領域に相当する所定数の枠要素をビューアに設定し、該各枠要素に、該当する位置の分割画像をそれぞれ当てはめて表示し、かつ、画像の相対移動に伴いビューアの表示領域から離れる分割画像を判断して枠要素への当てはめを解除し、ビューアの表示領域に近づく分割画像を判断して枠要素に新たに当てはめるものとすることができる。この場合、前記各枠要素に固有の識別情報を付与し、前記画像の相対移動に伴いビューアの表示領域から離れる分割画像の当てはめを解除する枠要素に、ビューアの表示領域に近づく分割画像を新たに当てはめるようにすることができる。また、前記各枠要素に固有の識別情報を付与し、ビューアは該固有の識別情報に、ビューアの表示領域における該枠要素の表示位置の情報と、該枠要素に当てはめる分割画像の識別情報を関連づけて保持し、ビューアは該情報に基づき、各枠要素に当てはめる分割画像を、ビューアの表示領域の所定位置に表示するものとすることができる。また、前記分割画像の識別情報は、画像全体におけるその番地に相当する情報をもって該画像全体における識別情報とし、ビューアは該分割画像の識別情報をもってサーバに該分割画像を要求し、サーバは該識別情報に該当する分割画像を判別してビューアに送信するものとすることができる。その場合、分割画像の識別情報は、各分割画像のファイル形式を識別する情報を有しないものとすることができる。また、前記所定数の枠要素全体でブロックを構成し、画像の相対移動が指示された時に、ビューアは、該ビューアの表示領域の原点に対する該ブロックの原点の移動すべき座標を演算し、該演算により求められたブロックの原点の座標に基づき、前記各枠要素の原点の移動すべき座標をそれぞれ演算し、該演算により求められた座標に各枠要素の原点をそれぞれ移動することにより、前記画像の相対移動を実現するものとすることができる。また、前記分割画像が前記画像を横方向に格子状に分割した画像でありかつ前記枠要素が横方向に所定数の連続し

た分割画像を当てはめるものであり、もしくは、前記分割画像が前記画像を縦方向に格子状に分割した画像でありかつ前記枠要素が縦方向に所定数の連続した分割画像を当てはめるものであり、または、前記分割画像が前記画像を横方向および縦方向に格子状に分割した画像でありかつ前記枠要素が横方向に所定数の連続した分割画像もしくは縦方向に所定数の連続した分割画像または横方向に所定数で縦方向に所定数の連続した分割画像を当てはめるもののものとすることができる。

#### 【0017】

また、同じ内容の画像を複数の倍率でかつ倍率ごとにそれぞれ複数の領域に分割してサーバから送信可能にし、ビューアから指示される倍率の画像について、この発明の画像表示方法を実行することができる。この場合、倍率にかかわらず、画像を横方向もしくは縦方向または横方向および縦方向に分割するピクセル数を同じとすることができる。また、1つの倍率で画像を表示している状態で該画像上の任意の位置にポインタを当てて倍率変更操作をした場合に、該ポインタが位置する画像上の位置を不動点として変更後の倍率の画像を表示するものとすることができる。

#### 【0018】

また、同じ内容の画像を複数の倍率でかつ倍率ごとにそれぞれ複数の領域に分割してサーバから送信可能にし、ビューアから指示される倍率の画像についてこの発明の画像表示方法を実行する場合には、倍率にかかわらず、前記枠要素の数を同じにすることができる。

#### 【0019】

また、この発明は、前記ビューアがWebブラウザであり、該Webブラウザでの各演算がサーバから送信されるHTMLに格納されているJavaScriptに基づき実行されるものとすることができる。これによれば、プラグインソフトウェアなしで画像表示を行うことができる。その場合、前記枠要素が、サーバから送信されるHTMLに記述されている<DIV>タグを使用して設定されているものとすることができる。また、サーバから送信されるHTMLは、分割画像ごとのファイル名及び／またはファイル形式等の属性の情報が組み込まれていないもの

とすることができる。

### 【0020】

#### 【発明の実施の形態】

この発明の実施の形態を以下説明する。この実施の形態では、プラグインソフトウェアなしで、新聞紙面画像を配信・閲覧する場合について説明する。図1にシステム構成を示す。Webサーバ（サーバコンピュータ）10とWebブラウザ（クライアントコンピュータ、ビューア）12とは、インターネット等の通信ネットワーク14を介して接続されている。Webサーバ10には、新聞紙面の画像データベースが予め格納されて用意されている。この画像データベースは、新聞1ページ全体を1つの画像とするもので、新聞を構成する各ページの画像を様々な倍率（長さ比で例えば、25%、33%、50%、67%（以上、実際の新聞紙面の縮小画像）、100%（実際の新聞紙面と等倍画像）、133%、200%、400%（以上、実際の新聞紙面の拡大画像））で（すなわち、同一ページの画像を複数の倍率で）保存している。各分割画像のファイル形式は同一である必要はなく、例えば分割画像ごとの特徴（写真か文字か、カラーか白黒か等）に応じて圧縮率が高いファイル形式（JPEG形式、PNG形式、GIF形式等）で保存する。したがって、新聞1ページの画像は、異なるファイル形式の分割画像が混在して構成される場合がある。このように、分割画像ごとに圧縮率が高いファイル形式を用いることにより送信時間を短縮できる。

### 【0021】

1つの倍率の1ページの画像は、図2に示すように、左上位置（画像の原点位置）から横方向（X軸方向）および縦方向（Y軸方向）にそれぞれ所定画素数ごとに格子状に分割されて、複数の矩形の分割画像の集合で構成されている。1つの分割画像を構成する画素数は、倍率の違いにかかわらず一定であり、例えば480×480ピクセルである（ただし、1ページ横方向全体の画素数が1つの分割画像について定められた横方向の画素数で割り切れない場合は、右端位置の分割画像の画素数は端数となる。同様に、1ページ縦方向全体の画素数が1つの分割画像について定められた縦方向の画素数で割り切れない場合は、下端位置の分割画像の画素数は端数となる。）。これに対し、1ページ分の紙面を構成する画

素数は、倍率に応じて変化する {倍率 (=長さ比) の二乗に比例する} ので、1 ページ分の紙面を構成する分割画像数は、倍率が低い画像ほど少なく、倍率が高い画像ほど多くなる。各分割画像には、1 ページの画像における各分割画像の位置 (番地) を示す座標値 (分割画像座標値) として、図 2 に示すように、横方向の座標を  $x$ 、縦方向の座標を  $y$  とし、左上角の分割画像位置を原点  $[0, 0]$  とし、右に行くに従い、また下に行くに従い 1 ずつ増加する分割画像座標値  $[x, y]$  に相当するファイル名がそれぞれ付与されて保存されている。1 ページの画像における右端の分割画像の横方向の座標値を  $x_1$ 、下端の分割画像の縦方向の座標値を  $y_1$  とすると、右下角の最後の分割画像の座標値は  $[x_1, y_1]$  となる。特定の新聞の、特定のページの、特定の倍率の、特定の分割画像は、新聞名、日付、朝/夕刊別、ページ番号、倍率、分割画像座標値  $[x, y]$  の各情報を組み合わせた識別情報で特定 (指定) される。新聞紙面を閲覧する場合は、Web ブラウザ 12 から、閲覧者により指示される閲覧位置に応じた該識別情報が送信され、Web サーバ 10 は該識別情報を解析して、該当する分割画像を送信し、Web ブラウザ 12 は該分割画像を受け取って、元の画像の並びに配列して表示する。上記のような識別情報を用いれば、Web サーバ 10 から最初に Web ブラウザ 12 に送信する HTML (後述する図 3 のステップ S2) に、分割画像ごとのファイル名 {ファイル形式を指定する識別子 (.jpg、.png、.gif 等) が付加されたデータ} を組み込んでおく必要がない (すなわち、Web ブラウザ 12 はファイル名やそのファイル形式を予め知らなくても、個々の分割画像を要求できる。) ので、該 HTML のデータ量は少なくて済む。

#### 【0022】

この実施例では、後述するように (図 6)、画像上にブラウザの表示領域全体が入る大きさの『ブロック』を設定する。このブロックは、例えば  $6 \times 6$  個の『セル』 (枠要素) で構成される。1 つのセルは 1 つの分割画像の大きさ (例えば  $480 \times 480$  ピクセル) に相当し、該当する位置の分割画像が当てはめ (割り当て) られる。ブロックは、画像が移動 (スクロール) しても常にブラウザの表示領域全体が入るように、ブラウザの表示領域に対する画像の移動に応じて、ブラウザの表示領域に対してセル単位で移動し、これに伴い、セルに当てはめられ

る分割画像も順次入れ替えられる。

#### 【0023】

新聞紙面を閲覧する場合のWebブラウザ12とWebサーバ10との間の通信のやり取りを図3に示す。閲覧者がWebブラウザ12からWebサーバ10に〇月〇日付の〇〇新聞の〇刊（朝刊または夕刊）を閲覧する旨の指示をすると、該当する内容のHTMLを要求する指示が出される（S1）。Webサーバ10はこの要求を受けて、図4に示す、分割画像データを当てはめる6×6個のセルを設定する<DIV>タグ、および、どの分割画像をどのセルに当てはめてブラウザ上のどの位置に表示するかを算出するためのJavaScriptを含んだHTMLをWebブラウザに返す（S2）。なお、6×6個のセルには、図4に示すように、「canvas0000」～「canvas0505」の名前が付けられている。「canvas」に続く4桁の数字のうち、最初の2桁はブロック内でのセルのX軸方向の番地（00～05）を示し、それに続く残りの2桁はブロック内でのセルのY軸方向の番地（00～05）を示す。

#### 【0024】

Webブラウザ12は、HTMLを受け取ると、該HTMLに記述されているJavaScriptに基づき、各セルにどの分割画像を当てはめるかおよび各セルに当てはめられた分割画像をブラウザ12上のどの位置に表示するかを算出する。そして、算出された分割画像がWebブラウザ12に既にダウンロードされてキャッシュメモリに保存されている分割画像であるかどうかを判断し、既に保存されている分割画像については、それを読み出して該当するセルに当てはめる。また、Webブラウザ12に保存されていない分割画像については、Webサーバ10に対し該当する分割画像を要求する（S3）。Webサーバ10は該要求に応じて、該当する分割画像データを送信する（S4）。Webブラウザ12は該分割画像データを受け取り、該当するセルに当てはめる。このようにして、6×6個のセルに分割画像がそれぞれ当てはめられて（関連づけられて）、前記算出された位置にそれぞれ表示される（S5）。これにより、Webブラウザ12の表示領域全体に、分割画像がつなぎ目なく表示されて、新聞紙面の画像が表示される。表示の一例を図5に示す（図5においてセルの境界線は仮想的に示したもので



、実際の画面では表示されない。) 。なお、閲覧開始当初は初期画面として、Webブラウザ12から閲覧の指示をした新聞の第1ページの所定倍率(紙面の記事全体が見渡せるように比較的低い倍率)の分割画像をWebブラウザ12がWebサーバ10に要求して表示するように、Webサーバ10から送信するHTMLが記述されている。

### 【0025】

#### 初期表示制御

ブロックを構成する各セルに当てはめる分割画像について説明する。ここでは、図6に示すように、ブラウザの表示領域16よりも大きい画像18(新聞紙面1ページ分の画像)を、ブラウザの表示領域16の中心と画像18の中心を一致させた状態に初期表示するものとする。また、ブロック20を6×6個のセル22で構成し、ブロック20の左側および上側を1セルずつ空けて(つまり、ブラウザの表示領域16の左上角が、ブロック20の左から2番目、上から2番目のセルに位置するように)表示するものとする。なお、以下の説明において、座標値あるいは番号(番地)はその種類に応じて次のように区別して表記する。

( , ) : ピクセル座標値(ピクセル単位の座標値)

[ , ] : 分割画像座標値(画像全体における分割画像ごとの座標値)

[ , ] : セル番号(ブロックにおけるセルごとの番号値)

ピクセル座標値は、ブラウザの原点(左上角)を原点(0, 0)とし、X軸方向の極性は右方向を+、左方向を-とし、Y軸方向の極性は下方向を+、上方向を-とする。

### 【0026】

図6において、各変数は次を意味する。

$X_i$  : 画像(新聞紙面1ページ)全体の幅(ピクセル値)

$Y_i$  : 画像(新聞紙面1ページ)全体の高さ(ピクセル値)

$X_b$  : ブラウザの幅(ピクセル値)

$Y_b$  : ブラウザの高さ(ピクセル値)

$X_1 = X_i / 2$  : 画像幅の半値 … (式1)

$Y_1 = Y_i / 2$  : 画像高さの半値 … (式2)

$$X_2 = X_b / 2 : \text{ブラウザ幅の半値} \quad \cdots (\text{式 } 3)$$

$$Y_2 = Y_b / 2 : \text{ブラウザ高さの半値} \quad \cdots (\text{式 } 4)$$

### 【 0 0 2 7 】

なお、画像全体の幅  $X_i$  および高さ  $Y_i$  の値は、WebサーバがHTMLにhiddenタグの属性値として埋め込んで送信する（図3のステップS2）ことにより、Webブラウザに与えられる。このhiddenタグの例を以下に示す。

```
<INPUT TYPE="hidden" NAME="imageWidth" VALUE="5986">
```

```
<INPUT TYPE="hidden" NAME="imageHeight" VALUE="8130">
```

この例によれば、 $X_i = 5986$ （ピクセル）、 $Y_i = 8130$ （ピクセル）が与えられる。また、ブラウザの幅および高さの値  $X_b$ 、 $Y_b$  は、Webサーバから送信されるHTML（図3のステップS2）に記述されているJavaScriptに基づきブラウザ自身で取得する。ブラウザの幅および高さの値  $X_b$ 、 $Y_b$  はJavaScriptの次の関数で求められる。

```
browserWidth=document.body.clientWidth; //表示されているブラウザの  
幅
```

```
browserHeight=document.body.clientHeight; //表示されているブラウザ  
の高さ
```

### 【 0 0 2 8 】

図6において、ブラウザの原点（左上角）を原点（0，0）とすると、画像の原点の座標値（ $X_0$ ， $Y_0$ ）は次式によって与えられる。

$$X_0 = X_2 - X_1 : \text{画像のX軸原点} \quad \cdots (\text{式 } 5)$$

$$Y_0 = Y_2 - Y_1 : \text{画像のY軸原点} \quad \cdots (\text{式 } 6)$$

1つの分割画像の大きさ（＝1つのセルの大きさ）を480×480ピクセルとすると、画像全体の分割画像数は次式によって与えられる。

$$G_{xi} = X_i / 480 : \text{X軸方向の全分割画像数} \quad \cdots (\text{式 } 7)$$

$$G_{yi} = Y_i / 480 : \text{Y軸方向の全分割画像数} \quad \cdots (\text{式 } 8)$$

この $G_{xi}$ 、 $G_{yi}$ が小数点以下の桁を有する場合は、小数点以下を切り上げた整数値が実際のX軸方向およびY軸方向の全分割画像数である。また、 $G_{xi}$ の小数点以下を切り捨てた値が右端の分割画像の横方向の座標値  $x_1$ （図2）であ

り、 $G_{y_i}$  の小数点以下を切り捨てた値が下端の分割画像の縦方向の座標値  $y_l$  (同) である。

### 【0029】

一方、ブラウザの表示領域 16 に必要な分割画像数は次式によって与えられる。

$$G_{x_b} = X_b / 480 : X \text{ 軸方向の表示に必要な分割画像数} \quad \cdots \text{ (式 9)}$$

$$G_{y_b} = Y_b / 480 : Y \text{ 軸方向の表示に必要な分割画像数} \quad \cdots \text{ (式 10)}$$

ブラウザの表示領域 16 に入る最初の分割画像 {ブラウザの原点 (0, 0) が属する (位置する) 分割画像} の座標  $[x_f, y_f]$  は、次式によって与えられる。

$$x_f = - \{ (X_0 / 480) \text{ の商の小数点以下を切り捨てた値} \} \quad \cdots \text{ (式 11)}$$

$$y_f = - \{ (Y_0 / 480) \text{ の商の小数点以下を切り捨てた値} \} \quad \cdots \text{ (式 12)}$$

### 【0030】

ブロックの左端に配置されるセル (X 軸方向のセル番号を  $S_{x_f}$  とする。以下「X 軸最初セル」という。) には、X 軸方向の分割画像座標値が  $x_f - 1$  の分割画像を当てはめる。同様に、ブロックの上端に配置されるセル (Y 軸方向のセル番号を  $S_{y_f}$  とする。以下「Y 軸最初セル」という。) には、Y 軸方向の分割画像座標値が  $y_f - 1$  の分割画像を当てはめる。すなわち、

$$S_{x_f} = x_f - 1 \quad \cdots \text{ (式 13)}$$

$$S_{y_f} = y_f - 1 \quad \cdots \text{ (式 14)}$$

とする。これによれば、ブロックの最初のセル  $[S_{x_f}, S_{y_f}]$  (ブロックの左上角に配置されるセル。以下「最初セル」という。) には、 $[x_f - 1, y_f - 1]$  の分割画像が当てはめられる。-1 を付加するのは、 $S_{x_f} = x_f$ 、 $S_{y_f} = y_f$  に設定すると、図 7 (a) に示すように、ブロック 20 の左端部分および上端部分がブラウザの表示領域 16 の左端部分および上端部分に接近して配置されることになり、画像を右下方向に移動 (スクロール) させた際に、新たに左側および上側からブラウザの表示領域 16 に入るべき分割画像が Web サーバか

ら新たに転送されて来るまでの待ち時間に、ブラウザの表示領域 16 の左端部分および上端部分に空白部分（画像が表示されない部分）が生じ易くなるためである。これに対し、-1 を付加すると、図 7 (b) に示すように、ブロック 20 は予めブラウザの表示領域 16 に対して 1 セル分以上左方向および上方向にオフセットして配置されることになり、画像を右下方向に移動させた際に、ブラウザの表示領域 16 の左端部分および上端部分に空白部分を生じにくくさせることができる。

### 【0031】

ブロックを構成するその他のセルに当てはめる分割画像の座標は、最初セル  $[S_{xf}, S_{yf}]$  に当てはめる分割画像の座標  $[x_f - 1, y_f - 1]$  に、X 軸方向並びに Y 軸方向に 1 ずつ順次足していくことにより求められる。すなわち、最初セル  $[S_{xf}, S_{yf}]$  を 0 番目のセルとして、X 軸方向に m 番目 ( $m = 0, 1, \dots, 5$ )、Y 軸方向に n 番目 ( $n = 0, 1, \dots, 5$ ) のセル  $[S_{xm}, S_{yn}]$  に当てはめられる分割画像座標値は、

$$S_{xm} = S_{xf} + m = x_f - 1 + m \quad \dots (式 15)$$

$$S_{yn} = S_{yf} + n = y_f - 1 + n \quad \dots (式 16)$$

で与えられる。したがって、ブロックの最後のセル  $[S_{x1}, S_{y1}]$ （ブロックの右下角に配置されるセル。以下「最後セル」という。なお、ブロックの右端に配置されるセルをそれぞれ「X 軸最後セル」、ブロックの下端に配置されるセルをそれぞれ「Y 軸最後セル」という。）に当てはめる分割画像の座標は、

$$S_{x1} = S_{xf} + 5 = x_f + 4 \quad \dots (式 17)$$

$$S_{y1} = S_{yf} + 5 = y_f + 4 \quad \dots (式 18)$$

となる。以上のようにして、ブロック 20 を構成する  $6 \times 6$  個のセル 22 に当てはめるべき分割画像が求められる。

### 【0032】

次に、ブロック 20 を構成する  $6 \times 6$  個のセル 22 のブラウザ上における表示位置について説明する。各セルのブラウザ上における表示位置は、該各セルの原点（左上角の位置）の座標値（ピクセル値）で特定（指定）される。各セルの原点の座標値は以下のようにして求められる。まず、最初セル  $[S_{xf}, S_{yf}]$

の原点 ( $X_{b0}$ ,  $Y_{b0}$ ) は、ブロックの原点に相当し (図6 参照)、

$$X_{b0} = X_0 + (S_{xf} \times 480) \quad \dots \text{(式19)}$$

$$Y_{b0} = Y_0 + (S_{yf} \times 480) \quad \dots \text{(式20)}$$

で与えられる。ブロックを構成するその他のセルの原点は、このようにして求められた最初セル [ $S_{xf}$ ,  $S_{yf}$ ] の原点 ( $X_{b0}$ ,  $Y_{b0}$ ) に X 軸方向並びに Y 軸方向に 480 ずつ順次足していくことにより求められる。すなわち、最初セル [ $S_{xf}$ ,  $S_{yf}$ ] を 0 番目のセルとして、X 軸方向に  $m$  番目 ( $m=0, 1, \dots, 5$ )、Y 軸方向に  $n$  番目 ( $n=0, 1, \dots, 5$ ) のセル [ $S_{xm}$ ,  $S_{yn}$ ] の原点 ( $X_{bm}$ ,  $Y_{bn}$ ) は、

$$X_{bm} = X_{b0} + (m \times 480) \quad \dots \text{(式21)}$$

$$Y_{bn} = Y_{b0} + (n \times 480) \quad \dots \text{(式22)}$$

で与えられる。

### 【0033】

Web サーバから送信される HTML (図3のステップ S2) には、以上の (式1) ~ (式22) に相当する演算を実行する JavaScript が記述されており、Web ブラウザは、該 HTML を受け取ると、該 JavaScript に基づき、ブロックを構成する  $6 \times 6$  個のセルに当てはめる分割画像座標値と、各セルのブラウザ上における表示位置 (各セルの原点位置を表示するピクセル座標位置) を算出する。そして、Web ブラウザは、算出された分割画像座標値を識別情報として Web サーバに該当する分割画像を要求する (Web サーバから既にダウンロードされてキャッシュメモリに保存されている分割画像についてはそれを使用する。) (図3のステップ S3)。Web サーバ 10 は該要求に応じて、該当する分割画像データを送信し (同 S4)、Web ブラウザ 12 は該分割画像データを受け取り、該当するセルに当てはめて、それぞれ前記求められた位置に表示する (同 S5)。

### 【0034】

ここで、(式1) ~ (式22) に実際の値を入れて、ブラウザによって求められる、各セルに当てはめる分割画像座標値と、各セルのブラウザ上における表示位置を確かめてみる。画像の大きさを X 軸方向: 5986 ピクセル、Y 軸方向:

8130ピクセルとし、ブラウザの幅をX軸方向:1020ピクセル、Y軸方向:556ピクセルとすると、 $X_i = 5986$ 、 $Y_i = 8130$ 、 $X_b = 1020$ 、 $Y_b = 556$ となる。(式1)、(式2)により、画像幅の半値 $X_1$ 、画像高さの半値 $Y_1$ は、

$$X_1 = 5986 / 2 = 2993$$

$$Y_1 = 8130 / 2 = 4065$$

となる。また、(式3)、(式4)により、ブラウザ幅の半値 $X_2$ 、ブラウザ高さの半値 $Y_2$ は、

$$X_2 = 1020 / 2 = 510$$

$$Y_2 = 556 / 2 = 278$$

となる。また、(式5)、(式6)により、画像の原点( $X_0$ ,  $Y_0$ )は、

$$X_0 = 510 - 2993 = -2483$$

$$Y_0 = 278 - 4065 = -3787$$

となる。

### 【0035】

1つの分割画像の大きさを480×480ピクセルとすると、(式7)、(式8)により、画像全体の分割画像数は、

$$G_{xi} = 5986 / 480 = 12.4708$$

$$G_{yi} = 8130 / 480 = 16.9375$$

となる。この $G_{xi} = 12.4708$ 、 $G_{yi} = 16.9375$ という数値は、画像全体の分割画像数が、X軸方向に13個、Y軸方向に17個で、全体として $13 \times 17 = 221$ 個となることを示す。また、 $G_{xi}$ の小数点以下を切り捨てた値が右端の分割画像の横方向の座標値 $x_1$ であり、 $G_{yi}$ の小数点以下を切り捨てた値が下端の分割画像の縦方向の座標値が $y_1$ である。すなわち、 $x_1 = 12$ 、 $y_1 = 16$ となる。

(式9)、(式10)により、ブラウザの表示領域に必要な分割画像数は、

$$G_{xb} = 1020 / 480 = 2.125$$

$$G_{yb} = 556 / 480 = 1.15833$$

となる。この $G_{xb} = 2.125$ 、 $G_{yb} = 1.15833$ という数値は、ブラ

ウザの表示領域全体に画像を表示するためには、ブロックを構成するセルが、X軸方向に少なくとも4個、Y軸方向に少なくとも3個必要であることを示す。この実施例では、画像の移動（スクロール）動作時に、空白部分（画像が表示されない部分）を生じにくくさせるために、前述のようにX軸方向が6個、Y軸方向が6個の合計 $6 \times 6 = 36$ 個のセルでブロックを構成している。ブロックを構成するセル数をより多くすれば、画像の移動動作時に空白部分をより生じにくくさせることができるが、その反面、該画像の移動動作時に各セルの座標計算に時間を要し、移動速度が遅くなる可能性がある（ただし、CPUの能力による）ので、移動動作時に空白部分が発生するのを防止することと画像の移動速度との兼ね合いで、ブロックを構成するセル数を設定する。

### 【0036】

（式11）、（式12）により、ブラウザの表示領域に入る最初の分割画像の座標 $[x_f, y_f]$ は、

$$x_f = - \{ (-2483 / 480) \text{の商の小数点以下を切り捨てた値} \} = 5$$

$$y_f = - \{ (-3787 / 480) \text{の商の小数点以下を切り捨てた値} \} = 7$$

となる。したがって、ブロックの最初セル $[S_{x_f}, S_{y_f}]$ に当てはめる分割画像の座標は、（式13）、（式14）により、

$$S_{x_f} = 5 - 1 = 4$$

$$S_{y_f} = 7 - 1 = 6$$

となる。また、ブロックの最後セル $[S_{x_l}, S_{y_l}]$ に当てはめる分割画像の座標は、（式17）、（式18）により、

$$S_{x_l} = S_{x_f} + 5 = 9$$

$$S_{y_l} = S_{y_f} + 5 = 11$$

となる。

### 【0037】

（式19）、（式20）により、最初セル $[S_{x_f}, S_{y_f}]$ の原点 $(X_{b0}, Y_{b0})$ は、

$$X_{b0} = -2483 + (4 \times 480) = -563$$

$$Y_{b0} = -3787 + (6 \times 480) = -907$$

となる。この最初セルの座標 ( $X_{b0}$ ,  $Y_{b0}$ ) を基に (式 21)、(式 22) により求められる各セルの原点の座標値を図 8 に示す。

#### 【0038】

ここで、ブロック 20 を構成する各セル 22 のセル番号の付与方法について説明する。このセル番号の付与も Web サーバから送信される HTML に記述されている JavaScript に基づき実行される。各セルのセル番号 [ $S_x$ ,  $S_y$ ] は、該各セルに当てはめられる分割画像の座標 [ $x$ ,  $y$ ] を、X 軸方向についてはブロックを構成する X 軸方向の総セル数、Y 軸方向についてはブロックを構成する Y 軸方向の総セル数でそれぞれ割った余りの値として付与する。これによれば、各分割画像が当てはめられるセルのセル番号 (各分割画像に割り当てられるセル番号) は、分割画像ごとに一意的に定まる。この実施例では、X 軸方向、Y 軸方向の総セル数はそれぞれ 6 であるから、セル番号は、X 軸方向、Y 軸方向ともに 0, 1, 2, ..., 5 の繰り返しとなる。この場合、各分割画像が当てはめられるセルのセル番号を図 9 に示す。図 9 の各分割画像位置に記述されている数値のうち、上段は各分割画像 24 の座標値を示し、下段は当てはめられるセルのセル番号を示す。セル番号が繰り返し使われても、ブロック内のセル数は  $6 \times 6$  個だけであるから、ブロック内で同時に同じセル番号が重複して使用されることはない。このようにセル番号を繰り返し用いることにより、図 4 に示すように、HTML に記述する、セルを設定するための  $\langle \text{DIV} \rangle$  タグは  $6 \times 6$  個だけ用意すれば足りることになる。

#### 【0039】

以上のようにして、Web サーバから送信される HTML に記述されている JavaScript に基づき、ブロック 20 を構成する  $6 \times 6$  個のセル 22 に当てはめる各分割画像 24 の座標値と該各セル 22 のセル番号および該各セル 22 のブラウザ上における表示位置 (各セルの原点位置を表示するピクセル座標位置) が求められたら、Web サーバに対し該当する分割画像 24 の送信を要求し (図 3 のステップ S3)、Web サーバから該要求に応じた分割画像データが送られてきたら (同 S4)、該当するセル番号に当てはめてブラウザに表示する (同 S5)。これにより、Web ブラウザ 12 の表示領域全体に、分割画像 24 がつなぎ目なく



表示されて、新聞紙面の画像 18 が表示される。

#### 【0040】

図 10 は、ブロック 20 を構成する各セル 22 に当てはめる各分割画像 24 の座標値と、該各セル 22 に付与するセル番号と、該各セル 22 のブラウザ上における表示位置（各セルの原点位置）の関係を求め、該当する分割画像を Web サーバに要求し、該要求に応じて Web サーバから送られてきた分割画像 24 を各セル 22 に当てはめるための JavaScript の関数（抜粋）を示す。この JavaScript の関数は、画像の左上角の分割画像 [0, 0] を始点として、分割画像位置を Y 軸方向に走査してブロックに含まれる分割画像の有無を判断し、ブロックに含まれる分割画像があった場合は、該分割画像が当てはめられるセルに、該分割画像の X 軸方向の座標値  $x$  を 6 で割った余りを X 軸方向のセル番号  $sellX$  として付与し、該分割画像の Y 軸方向の座標値  $y$  を 6 で割った余りを Y 軸方向のセル番号  $sellY$  として付与する。そして、「document.all("canvas" +  $sellX$  +  $sellY$ ).innerHTML = "<IMG ID='image' +  $x$  +  $y$  + " 'SRC='image.cgi?a=xxx&b=xxx&c=xxx&d=xxx&e=xxx&f=xxx&g=xxx' STYLE='left:" +  $originX$  + ";top:" +  $originY$  + ";'>";」により、該当する分割画像を Web サーバに要求し、該要求に応じて Web サーバから送られてきた分割画像を該当するセルに当てはめる。なお、上の JavaScript の関数は、「canvas  $sellX$   $sellY$ 」（ $sellX=00\sim05$ 、 $sellY=00\sim05$ ）の ID を持つ DIV タグ（図 4 参照）を、このセルに当てはめる分割画像の座標値「image  $x$   $y$ 」の ID を持つ IMG タグに変換し、該 IMG タグのソースとして「image.cgi」（Web サーバ側の画像データ検索および転送モジュール）からの分割画像データを埋め込むことを意味する。そのとき「a=xxx&b=xxx&c=xxx&d=xxx&e=xxx&f=xxx&g=xxx」は、Web サーバに保存されている画像を特定するためのパラメータ（新聞名、日付、朝／夕刊別、ページ番号、倍率等の情報）として、特定の画像ファイルを検索するための値となる。「STYLE='left:" +  $originX$  + ";top:" +  $originY$ 」はこのセル（分割画像）の原点（左上の位置）を表示するブラウザの表示領域上の座標位置（ピクセル値）を示す。

#### 【0041】

以上の処理をその列の Y 軸方向の最後の分割画像まで行ったら、その 1 つ右隣

の列に移って同様に上端からブロックに含まれる分割画像の有無を判断し、ブロックに含まれる分割画像が当てはめられるセルに、X軸方向のセル番号sellXおよびY軸方向のセル番号sellYを付与する。そして、X軸方向の最後の列の最後の分割画像まで達したら、処理を終了する。

#### 【0042】

##### スクロール制御

次に、以上のようにして初期表示がなされた後に、閲覧者のスクロール操作によりブラウザの表示画面上で画像を任意の位置に移動させる（つまり、閲覧位置を移動する）方法について説明する。図11は、スクロール操作による画像の移動動作を示す。図11では、画像18をX軸方向の右方向に $X_d$ （ピクセル値）、Y軸方向の下方向に $Y_d$ （ピクセル値）移動させた場合（移動後の画像位置を符号18'で示す。）を示す。ブロック20は画像18と一体に移動するが、この移動によってブロック20はブラウザの表示領域16に対し相対移動し、何も手当をしなければ、ブロック20はいずれブラウザの表示領域16から外れてしまう。そこで、移動量に応じて画像18に対するブロック20の位置を変更し、これに伴い該ブロック20を構成する各セル22に対する分割画像の当てはめを更新する制御（ブロックの再設定）を併せて行う。このブロックの再設定により、ブロック20は常にブラウザの表示領域16全体を含む（ブラウザの表示領域16全体が常にブロック20内に含まれる）ように制御される。図11の例では、スクロール操作によりブロック20がブラウザの表示領域16に対し相対的に右方向に移動して、セルの境界線が1本ブラウザの原点を越えているので、ブロックの再設定により、右端の1列のセルが削除され、その分左端に1列のセルが追加されている。それらの間の5列のセルには変更はない。

#### 【0043】

スクロール操作時の制御フローを図12に示す。図12の制御は、Webサーバから送信されるHTMLに記述されているJavaScriptによって行われる。スクロール操作時の制御は、画像を移動させる制御と、ブロックの再設定制御で構成される。はじめに、画像の移動制御について説明する。新聞紙面の画像が表示されている画面上の任意の位置にマウスカーソル（ポインタ）を当ててマウス左ボ

タンを押下し（マウスダウン操作）（図12のステップS11）、押下したままマウスカーソルを移動させると（ドラッグ操作）、マウスカーソルに追従して画像が移動する（同S12）。この場合、セル単位で移動量の演算をすると、演算処理に時間がかかり、移動動作が遅くなる。そこで、図4のHTMLのように、 $6 \times 6$ 個のセルを設定する<DIV>タグ全体を、名前がcanvas BLOCKである<DIV>タグで1つのブロックとして束ねて、該1つのブロックについて移動量を演算して該ブロックの移動後の原点を求め、該求められたブロックの原点の座標に基づき、該ブロックを構成する各セルの原点の座標を演算して求める。これによれば、ブロックの原点に対する各セルの原点の相対位置は予め定まっており、簡単な演算で求めることができ、一方複雑な演算を要する移動量の演算は1つのブロックについてのみ行えばよいので、セル単位で（セルごとに）移動量の演算をする場合に比べて全体として演算量が少なくて済み、移動動作を高速化することができる。

#### 【0044】

図13は、ブロック（画像）の移動量（＝マウスカーソルの移動量）を計算してブロックの原点の移動すべき座標を求めるためのJavaScriptの関数を示す。この関数によりブロックの原点の移動すべき座標が求められたら、この座標がブロックの最初セル $[S_{xf}, S_{yf}]$ の原点の座標になるので、他のセルについてもX軸方向およびY軸方向にそれぞれ480ずつ順次足していくことによりそれぞれの原点の座標を求める。このような演算をマウスカーソルが移動されている間中繰り返すことにより、マウスカーソルに追従して画像を移動させることができる。マウスカーソルを任意の位置に移動後、マウスボタンを離す（マウスアップ操作をする）と、画像はその位置で停止する。

#### 【0045】

次に、ブロックの再設定制御について説明する。ドラッグ操作を終了して、マウスアップ操作をすると、移動後の画像上でブラウザの原点（0，0）が属するセル位置（分割画像位置）が判断され、ドラッグ前の位置に対して変化しているかどうか（つまり、ブラウザの原点（0，0）がセルの境界を超えたかどうか）が判断される（図12のステップS13）。そして、変化している場合（セルの

境界を越えた場合)は、その変化した分(セルの境界を越えた本数分)ブロック位置を画像の移動と同じ方向に移動させるとともに、該ブロックを構成する各セルに対する分割画像の当てはめを更新することによりブロックの再設定を行う(同S14)。

#### 【0046】

ブロックの再設定制御は次の演算により実現される。ドラッグ操作による画像の移動量を図11のようにX軸方向の右方向に $X_d$ 、Y軸方向の下方向に $Y_d$ とすると、画像の移動量 $X_d$ 、 $Y_d$ は、ドラッグ終了位置(マウスアップ操作位置)でのマウスカーソル座標値からドラッグ開始位置(マウスダウン操作位置)でのマウスカーソル座標値を引算することにより得られる。移動後の画像の原点の座標値( $X_{d0}$ ,  $Y_{d0}$ )は、次式によって与えられる。

$$X_{d0} = X_{f0} + X_d : \text{移動後の画像のX軸原点} \cdots (\text{式23})$$

$$Y_{d0} = Y_{f0} + Y_d : \text{移動後の画像のY軸原点} \cdots (\text{式24})$$

ただし、 $X_{f0}$ : 前回のスクロール操作による画像の最終的なX軸原点。初回のスクロール操作であれば、 $X_{f0} = X_0$  (初期表示時の画像のX軸原点)である。

$Y_{f0}$ : 前回のスクロール操作による画像の最終的なY軸原点。初回のスクロール操作であれば、 $Y_{f0} = Y_0$  (初期表示時の画像のY軸原点)である。

#### 【0047】

また、移動後にブラウザの表示領域に入る最初の分割画像 {ブラウザの原点(0, 0)が属する分割画像} の座標  $[x_{df}, y_{df}]$  は、次式によって与えられる。

$$x_{df} = - \{ (X_{d0} / 480) \text{の商の小数点以下を切り捨てた値} \} \cdots (\text{式25})$$

$$y_{df} = - \{ (Y_{d0} / 480) \text{の商の小数点以下を切り捨てた値} \} \cdots (\text{式26})$$

したがって、ドラッグによってセルの境界がブラウザの原点(0, 0)を越えた数 $S_{xd}$ ,  $S_{yd}$ は、次式によって与えられる。

$$S_{x d} = x_{f 0} - x_{d f} \quad \cdots \text{(式 27)}$$

$$S_{y d} = y_{f 0} - y_{d f} \quad \cdots \text{(式 28)}$$

ただし、 $x_{f 0}$ ：前回のブロックの再設定時のブラウザの表示領域に入る最初の分割画像のX軸座標

$y_{f 0}$ ：前回のブロックの再設定時のブラウザの表示領域に入る最初の分割画像のY軸座標

#### 【0048】

(式 27) によって求められる  $S_{x d}$  が 0 であれば、セルの境界はブラウザの原点 (0, 0) をX軸方向に越えてないことがわかり、+の値であれば、その本数分のセルの境界がブラウザの原点 (0, 0) を右方向に越えたことがわかり、-の値であれば、その本数分のセルの境界がブラウザの原点 (0, 0) を左方向に越えたことがわかる。また、(式 28) によって求められる  $S_{y d}$  が 0 であれば、セルの境界はブラウザの原点 (0, 0) をY軸方向に越えてないことがわかり、+の値であれば、その本数分のセルの境界がブラウザの原点 (0, 0) を下方向に越えたことがわかり、-の値であれば、その本数分のセルの境界がブラウザの原点 (0, 0) を上方向に越えたことがわかる。

#### 【0049】

(式 27)、(式 28) の演算の結果、 $S_{x d}$ 、 $S_{y d}$  がいずれも 0 の場合は、ブラウザの原点 (0, 0) が属する分割画像の位置はドラッグの前後で変化がないことになるから、ブロックの再設定は行われず。これに対し、(式 27)、(式 28) の演算の結果、 $S_{x d}$ 、 $S_{y d}$  のうち少なくとも一方が 0 以外の場合は、ブラウザの原点 (0, 0) が属する分割画像の位置がドラッグの前後で変化したことになるから、ブロックの再設定が行われる。

#### 【0050】

ブロックの再設定は、 $S_{x d}$ 、 $S_{y d}$  に相当するセル数分、ブロック全体を、画像に対し、画像の移動方向と逆方向に相対的に移動させることで実現される。このブロックの再設定によって、該ブロックを構成する各セルには次のように分割画像が当てはめられる。ブロックの左端に配置されるセル (X軸最初セル)  $S_{x d f}$  には、X軸方向の分割画像座標値が  $x_{d f} - 1$  の分割画像が当てはめられ

る。同様に、ブロックの上端に配置されるセル（Y軸最初セル）には、Y軸方向の分割画像座標値が  $y_{df} - 1$  の分割画像が当てはめられる。すなわち、

$$S_{xdf} = x_{df} - 1 \quad \cdots \text{(式 29)}$$

$$S_{ydf} = y_{df} - 1 \quad \cdots \text{(式 30)}$$

とする。これによれば、再設定後のブロックの左上角に配置されるセル（最初セル） $[S_{xdf}, S_{ydf}]$  には、 $[x_{df} - 1, y_{df} - 1]$  の分割画像が当てはめられる。-1を付加するのは、(式13)、(式14)で-1を付加したのと同じ目的である。すなわち、ブロックをブラウザの表示領域に対して1セル分以上左方向および上方向にオフセットして配置することにより、画像を右下方向に移動させた際に、ブラウザの表示領域の左端部分および上端部分に空白部分を生じにくくする。

#### 【0051】

ブロックを構成するその他のセルに当てはめる分割画像の座標は、最初セル $[S_{xdf}, S_{ydf}]$ に当てはめる分割画像の座標 $[x_{df} - 1, y_{df} - 1]$ に、X軸方向並びにY軸方向に1ずつ順次足していくことにより求められる。すなわち、最初セル $[S_{xdf}, S_{ydf}]$ を0番目のセルとして、X軸方向にm番目（ $m=0, 1, \dots, 5$ ）、Y軸方向にn番目（ $n=0, 1, \dots, 5$ ）のセル $[S_{xdm}, S_{ydn}]$ に当てはめられる分割画像座標値は、

$$S_{xdm} = S_{xdf} + m = x_{df} - 1 + m \quad \cdots \text{(式 31)}$$

$$S_{ydn} = S_{ydf} + n = y_{df} - 1 + n \quad \cdots \text{(式 32)}$$

で与えられる。したがって、ブロックの右下角に配置されるセル（最後セル） $[S_{xdl}, S_{ydl}]$ に当てはめる分割画像の座標は、

$$S_{xdl} = S_{xdf} + 5 = x_{df} + 4 \quad \cdots \text{(式 33)}$$

$$S_{ydl} = S_{ydf} + 5 = y_{df} + 4 \quad \cdots \text{(式 34)}$$

となる。

#### 【0052】

以上のようにして、再設定されたブロックを構成する $6 \times 6$ 個のセルに当てはめるべき分割画像が求められる。 $6 \times 6$ 個のセルに当てはめるべき分割画像が求められたら、新たにブロックの領域に入ってきた分割画像について、未取得の場

合はWebサーバに要求し、Webサーバから既にダウンロードされてキャッシュメモリに保存されている分割画像についてはそれを読み出して、それぞれ該当するセルに当てはめて表示する。なお、図9に示したように、各分割画像が当てはめられるセルのセル番号（各分割画像に割り当てられるセル番号）は、分割画像ごとに一意的に定まっている。前回のブロックの再設定からブロック内にとどまっている（スクロール操作によってもブロック内から出なかった）分割画像については、セル番号の変更はない。

### 【0053】

以上のブロックの再設定制御による、ブロックを構成する各セルに対する分割画像の当てはめの変更の一例を図14に示す。図14の各セル位置に記述されている数値のうち、上段は分割画像座標値を示し、下段はセル番号を示す。この例は、スクロール操作による画像の移動によって、セルの境界がブラウザの原点（0，0）をX軸方向の右方向に1本だけ越えて、Y軸方向には1本も越えていない場合（図11に示すように画像が移動した場合）を示す。

### 【0054】

#### 表示倍率変更制御

新聞を閲覧中に表示倍率を変更する場合の制御について説明する。この表示倍率変更制御は、マウスカーソル（ポインタ）位置を不動点として表示倍率に変更されるように（つまり、マウスカーソルが位置する紙面上の位置が倍率変更前と倍率変更後で変化しないように）制御している。図15はその制御フローを示す。図15の制御は、Webサーバから送信されるHTMLに記述されているJavaScriptによって行われる。新聞を閲覧中に、表示されている新聞紙面の画像上の任意の位置（表示倍率を変更して閲覧したい記事位置）にマウスカーソルを当てて（S21）、マウスを右クリック操作すると（S22）、その時のマウスカーソル（ポインタ）の座標値 {ブラウザの原点（0，0）を原点とするピクセル値} が取得される（S23）。なお、画像の原点の座標値（ $X_{d0}$ ， $Y_{d0}$ ）は、初期値（ $X_0$ ， $Y_0$ ）は（式5）、（式6）によって与えられ、その後はスクロール操作がされるごとに（式23）、（式24）によって更新されるので、表示倍率変更操作時の画像の原点の座標値（ $X_{d0}$ ， $Y_{d0}$ ）は、既に取得されてい

る。

### 【0055】

前記マウスの右クリック操作により表示倍率を選択するためのポップアップメニューが表示される（S24）。閲覧者がポップアップメニューに表示された表示倍率の数値（長さ比）の中から任意の倍率を選択する操作をすると（S25）、前記右クリックした時のマウスカーソル位置を不動点として、選択した倍率で画像を表示するために必要な、該画像の原点位置を求めるための演算が実行される（S26）。その演算内容について図16を参照して説明する。図16の例は、表示倍率変更前に表示されていた画像1（18-1）を、表示倍率変更操作位置（右クリックした時のマウスカーソル位置。以下「処理中心」と言う。）を不動点として画像2（18-2）に拡大表示する場合について示したものである。

### 【0056】

図16において、各変数は次を意味する。

$X_m$  : ブラウザの原点と処理中心とのX軸方向距離

$X_{m11}$  : ブラウザの原点と画像1の原点とのX軸方向距離

$X_{m12}$  : 画像1の原点と処理中心とのX軸方向距離

$X_{m21}$  : ブラウザの原点と画像2の原点とのX軸方向距離

$X_{m22}$  : 画像2の原点と処理中心とのX軸方向距離

$Y_m$  : ブラウザの原点と処理中心とのY軸方向距離

$Y_{m11}$  : ブラウザの原点と画像1の原点とのY軸方向距離

$Y_{m12}$  : 画像1の原点と処理中心とのY軸方向距離

$Y_{m21}$  : ブラウザの原点と画像2の原点とのY軸方向距離

$Y_{m22}$  : 画像2の原点と処理中心とのY軸方向距離

画像1の表示倍率に対する画像2の表示倍率の比を $\alpha$ （ $\alpha > 1$ ）とすると、

$$X_{m22} = \alpha \cdot X_{m12} \quad \cdots \text{(式35)}$$

$$Y_{m22} = \alpha \cdot Y_{m12} \quad \cdots \text{(式36)}$$

が成り立つ。また、図16から

$$X_{m12} = X_m + X_{m11} \quad \cdots \text{(式37)}$$

$$Y_{m12} = Y_m + Y_{m11} \quad \cdots \text{(式38)}$$



である。(式37)、(式38)を(式35)、(式36)に代入すると、

$$X_{m22} = \alpha \cdot X_m + \alpha \cdot X_{m11} \quad \cdots (\text{式} 39)$$

$$Y_{m22} = \alpha \cdot Y_m + \alpha \cdot Y_{m11} \quad \cdots (\text{式} 40)$$

となる。

#### 【0057】

図16から

$$X_{m21} = X_{m22} - X_m \quad \cdots (\text{式} 41)$$

$$Y_{m21} = Y_{m22} - Y_m \quad \cdots (\text{式} 42)$$

である。(式39)、(式40)を(式41)、(式42)に代入すると、

$$X_{m21} = (\alpha \cdot X_m + \alpha \cdot X_{m11}) - X_m = (\alpha - 1) X_m + \alpha \cdot X_{m11} \quad \cdots (\text{式} 43)$$

$$Y_{m21} = (\alpha \cdot Y_m + \alpha \cdot Y_{m11}) - Y_m = (\alpha - 1) Y_m + \alpha \cdot Y_{m11} \quad \cdots (\text{式} 44)$$

となる。(式43)、(式44)で求められる座標値( $X_{m21}$ ,  $Y_{m21}$ ) (ピクセル値)が、倍率変更後の画像の原点位置である。

#### 【0058】

図15のステップS26で、以上の演算により倍率変更後の画像の原点位置が求められたら、ブロックを構成する $6 \times 6$ 個のセルに当てはめる分割画像を算出する(S27)。なお、1つのセルに当てはめる1つの分割画像を構成する画素数は、前述のように倍率の違いにかかわらず一定であり、例えば $480 \times 480$ ピクセルである。各セルに当てはめる分割画像は、前記(式11)～(式16)に基づいて算出することができる。すなわち、(式11)、(式12)からブラウザの表示領域に入る最初の分割画像 {ブラウザの原点(0, 0)が属する分割画像} の座標 $[x_f, y_f]$ を求める(ただし、(式11)、(式12)は、画像の原点の初期値 $X_0$ 、 $Y_0$ に代えて、(式43)、(式44)で求められた倍率変更後の画像の原点位置 $X_{m21}$ 、 $Y_{m21}$ を用いる。))。

#### 【0059】

ブラウザの表示領域に入る最初の分割画像の座標 $[x_f, y_f]$ が求まったら、前記(式13)、(式14)により、ブロックの左端に配置されるセル $S_{x_f}$

(X軸最初セル)に当てはめる分割画像座標値として $x_f - 1$ を求める。同様に、ブロックの上端に配置されるセル $S_{y_f}$ (Y軸最初セル)に当てはめる分割画像座標値として $y_f - 1$ を求める。さらに、最初セル $[S_{x_f}, S_{y_f}]$ を0番目のセルとして、前記(式15)、(式16)により、X軸方向に $m$ 番目( $m = 0, 1, \dots, 5$ )、Y軸方向に $n$ 番目( $n = 0, 1, \dots, 5$ )のセル $[S_{x_m}, S_{y_n}]$ に当てはめられる分割画像座標値として、 $[x_f - 1 + m, y_f - 1 + n]$ をそれぞれ求める。

#### 【0060】

図15のステップS27で、以上の演算により、ブロックを構成する $6 \times 6$ 個のセルに当てはめる分割画像が算出されたら、前記(式19)～(式22)により、ブロックを構成する $6 \times 6$ 個のセルのブラウザ上における表示位置を求める(ただし、画像の原点の初期値 $X_0$ 、 $Y_0$ に代えて、倍率変更後の画像の原点位置 $X_{m21}$ 、 $Y_{m21}$ を用いる。)(S28)。すなわち、(式19)、(式22)によりブロックの原点の座標を求め、さらに、(式21)、(式22)により各セルの原点の座標を求める。

#### 【0061】

以上のようにして、倍率変更後の $6 \times 6$ 個のセルに当てはめるべき分割画像およびその表示位置が求められたら、未取得の分割画像についてはWebサーバに要求し(S29)、Webサーバから既にダウンロードされてキャッシュメモリに保存されている分割画像についてはそれを読み出して、それぞれ該当するセルに当てはめて表示する(S30)。

#### 【0062】

以上のようにして、処理中心を不動点として表示倍率に変更される。なお、以上は画像を拡大表示する場合について説明したが、縮小表示する場合も、表示倍率の比 $\alpha$ が $\alpha < 1$ に変わるだけで、処理内容は拡大表示する場合と同じである。

#### 【0063】

##### ページめくり制御

新聞紙面を閲覧している際に、ブラウザ画面上のツールバー等で所定のページめくり操作をすると、Webブラウザは、HTMLに記述されているJavaScript

に基づき、指示されたページについて、所定倍率（紙面の記事全体が見渡せるように比較的低い倍率）の分割画像をWebサーバに要求し（Webサーバから既にダウンロードされてキャッシュメモリに保存されている分割画像についてはそれを読み出して用いる。）、前記閲覧開始当初の初期画面と同様に、ブラウザの表示領域の中心と画像の中心を一致させた状態に該画像を初期表示する。該初期表示後のスクロール制御、倍率変更制御は前述と同じに実行される。

#### 【0064】

なお、以上説明した実施の形態では、画像を縦横両方向に分割したが、縦横一方にのみ分割することもできる。また、セル（枠要素）を縦横両方向に配列したが、縦横一方にのみ配列することもできる。また、分割画像およびセルを正方形としたが、長方形とすることもできる。分割画像およびセルを矩形以外の形状に分割することもできる。また、ブロックを構成するセル数を縦横同一としたが、縦横非同一とすることもできる。また、ブロックをブラウザの表示領域に対して予め1セル分左方向および上方向にオフセットさせが、オフセット量は2セル以上にすることもできる。また、分割画像をサーバに予め用意してビューアからの要求を待って送信するようにしたが、ビューアからの要求を待って分割画像を生成して送信することもできる。また、初期表示した後、スクロール操作を待ってその周囲の分割画像をダウンロードするようにしたが、初期表示した後、スクロール操作を待たずにその周囲の分割画像を順次ダウンロードすることもできる。また、新聞を閲覧する場合について説明したが、その他の画像（例えば漫画）を閲覧する場合にもこの発明を適用できる。また、Webブラウザを使用して閲覧する場合について説明したが、プラグインソフトウェアを使用して閲覧する場合にも、この発明を適用できる。

#### 【図面の簡単な説明】

【図1】 この発明の実施の形態を示すシステム構成図である。

【図2】 画像分割の説明図である。

【図3】 図1のシステムにおいて新聞紙面を閲覧する場合のWebブラウザ12とWebサーバ10との間の通信のやり取りを示す図である。

【図4】 分割画像データを当てはめる6×6個のセルをブロック化するた

めのHTMLソースプログラムである。

【図5】 図1のWebブラウザ12に表示される新聞紙面の画像の一例を示す図である。

【図6】 新聞紙面1ページの画像とブラウザの表示領域およびブロックの位置関係の一例を示す図である。

【図7】 ブロックとブラウザの表示領域の関係を示す図で、(a)はオフセットなしの場合、(b)は-1セル分のオフセットありの場合である。

【図8】 ブロックを構成する各セルの原点の座標値の一例を示す図である。

【図9】 新聞1ページの画像を構成する各分割画像の座標値と、該各分割画像が当てはめられるセルのセル番号を示す図である。

【図10】 ブロックを構成する各セルに当てはめる分割画像座標値と、該各セルに付与するセル番号と、該各セルのブラウザ上における表示位置の関係を求め、該当する分割画像をWebサーバに要求し、該要求に応じてWebサーバから送られてきた分割画像を各セル当てはめるためのJavaScriptの関数(抜粋)を示す図である。

【図11】 スクロール操作による画像の移動動作の一例を示す図である。

【図12】 スクロール操作時の制御内容を示すフローチャートである。

【図13】 ブロックの移動量を計算してブロックの原点の移動すべき座標を求めるためのJavaScriptの関数を示す図である。

【図14】 ブロックの再設定制御による、ブロックを構成する各セルに対する分割画像の当てはめの変更の一例を示す

【図15】 表示倍率変更制御の制御内容を示すフローチャートである。

【図16】 図15のステップS26で実行する演算の内容を説明するための図である。

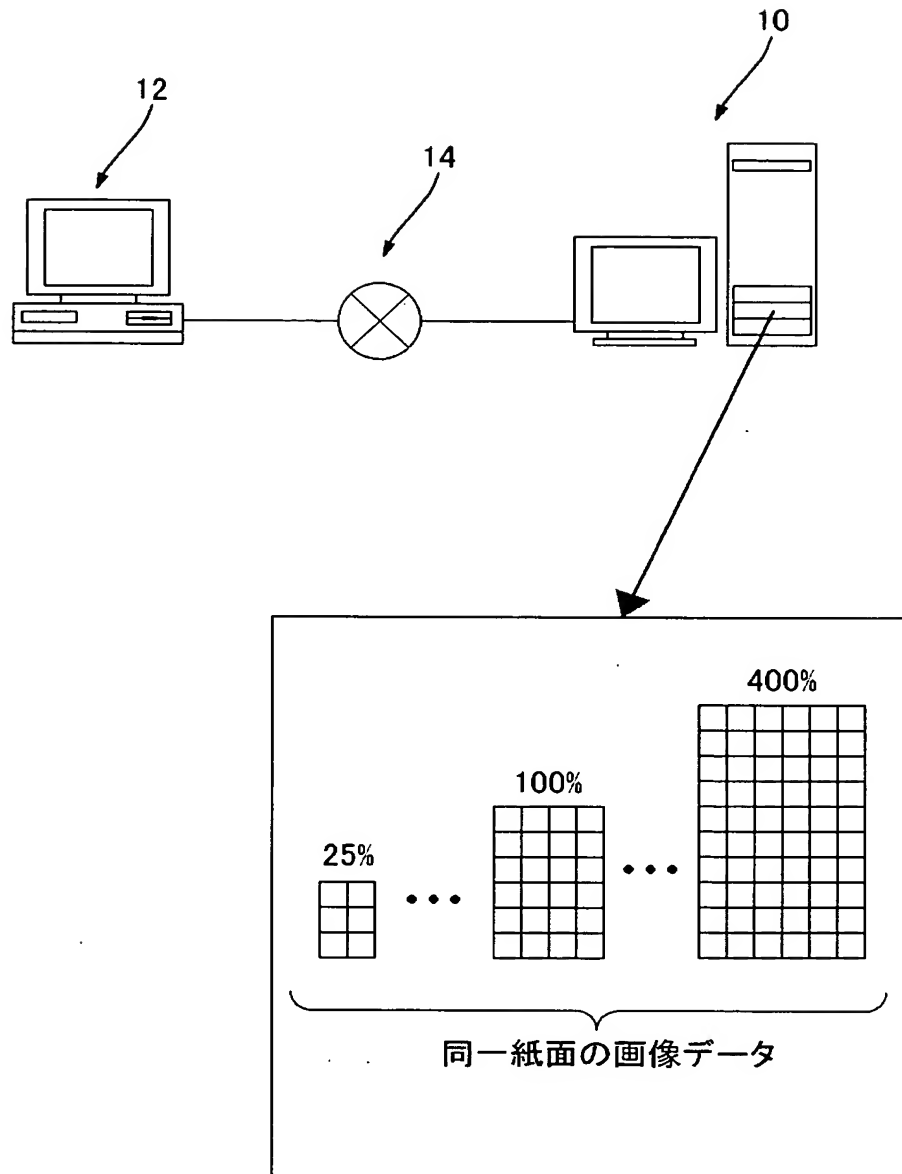
#### 【符号の説明】

10…Webサーバ(サーバコンピュータ)、12…Webブラウザ(クライアントコンピュータ、ビューア)、14…通信ネットワーク、16…ブラウザの表示領域(ビューアの表示領域)、18、18-1、18-2…画像、20…ブ

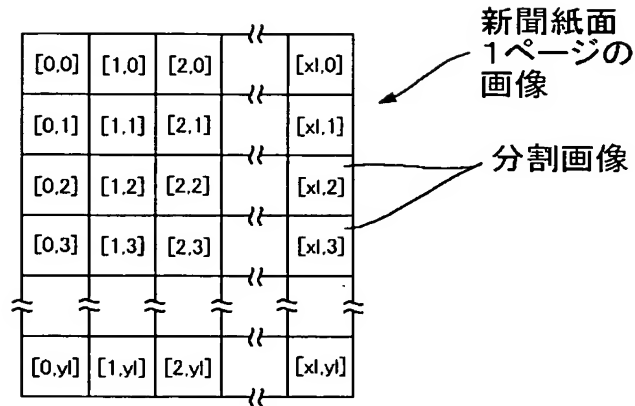
ロック、 2 2 …セル（枠要素）、 2 4 …分割画像

【書類名】 図面

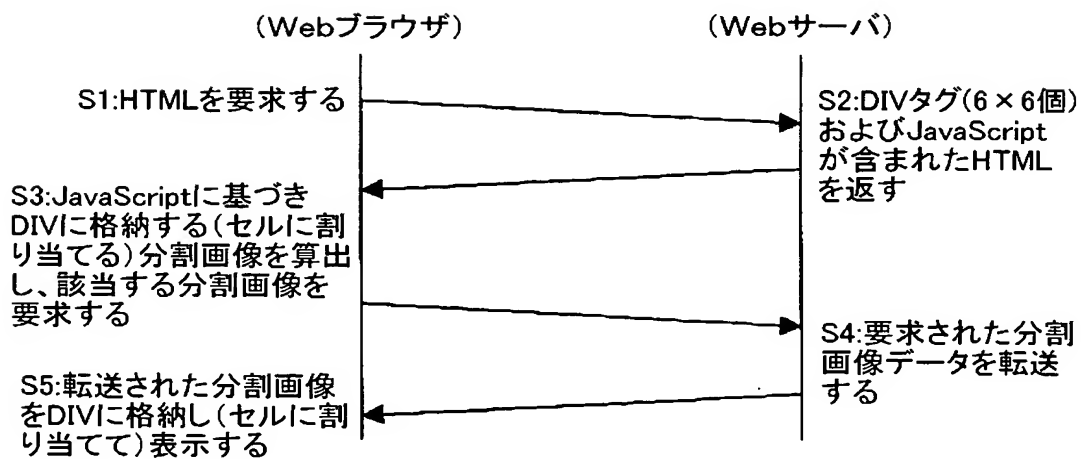
【図 1】



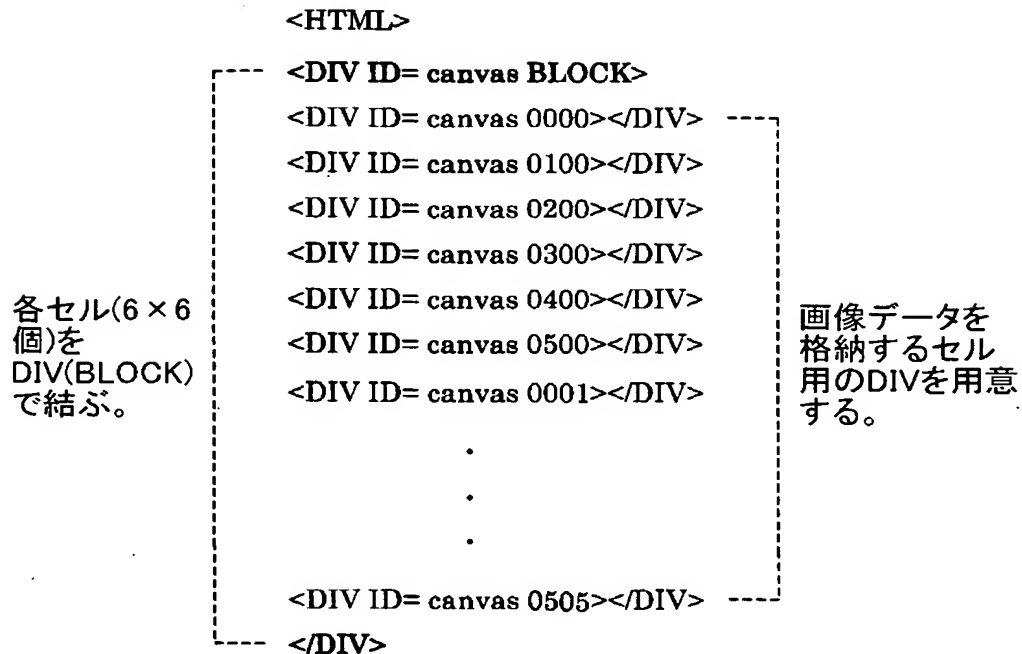
【図 2】



【図 3】



【図 4】



【図 5】

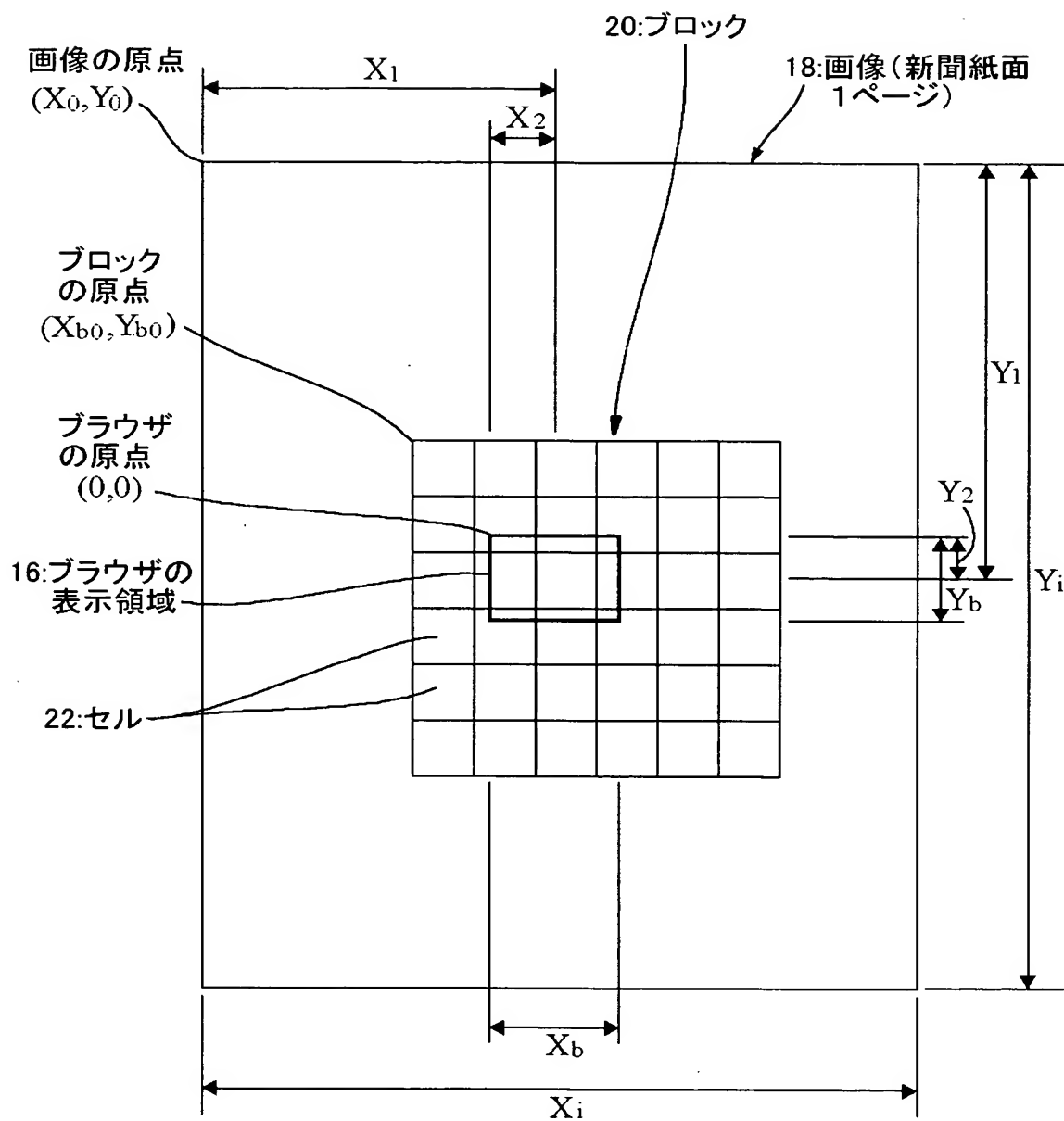
ブラウザの原点(0,0)

セルの境界線

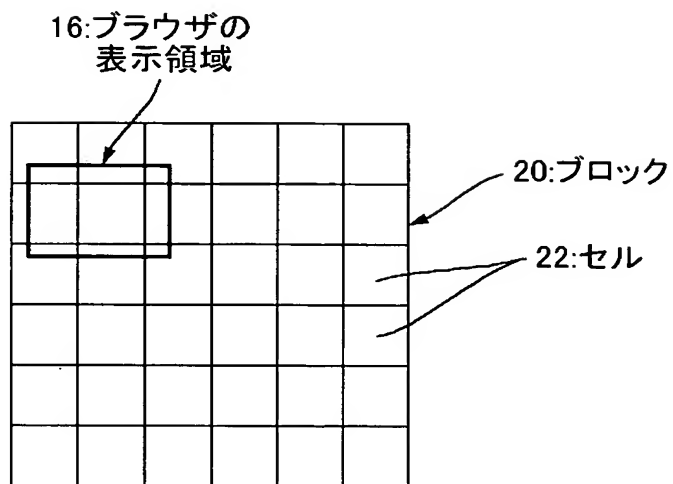
銘柄	始値	高値	安値	終値	比	出	養命酒	770	770	760
・三井物産	117	119	116	119	△	2	・三国コカ	860	888	854
・日水	257	257	252	257	△	1	・近畿コカ	680	680	676
・マルハ	126	130	125	130	△	5	X四国コカ	1010	1034	1008
Xサカタネ	1263	1263	1250	1263	△	1	Xコカウエスト	2000	2000	1981
Xホクト	1850	1850	1810	1813	▽	48	Yコカセトラ	6490	6490	6350
・三井山	50	52	49	52	△	8	ADYDO	2075	2085	2045
S住友炭	38	39	37	39	△	1	・カルピス	616	630	616
・日鉄鉱	226	228	209	209	▽	22	ポツカ	307	315	300
・三井松島	125	131	121	130	△	9	X伊藤園	3920	3950	3870
・帝石	403	407	390	400	▽	6	Xキーコーヒー	1495	1495	1475
・ガス開	529	530	516	525	▽	9	Xキリンビール	2055	2070	2045
・太平発	50	51	49	50	0	0	AJニカフェ	1478	1485	1470
							Gアサヒ飲料	498	500	485
							・日清イリオ	320	321	315
							X不二油	971	975	953
							・Jオイル	178	184	173
							・キッコマン	784	785	760
							・味の素	1257	1257	1233
							V&Vコービー	014	021	007



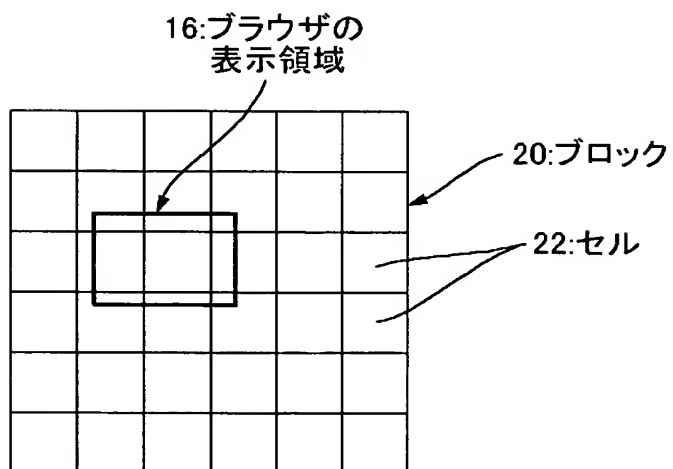
【図 6】



【図 7】



(a) オフセット無し



(b) オフセット有り

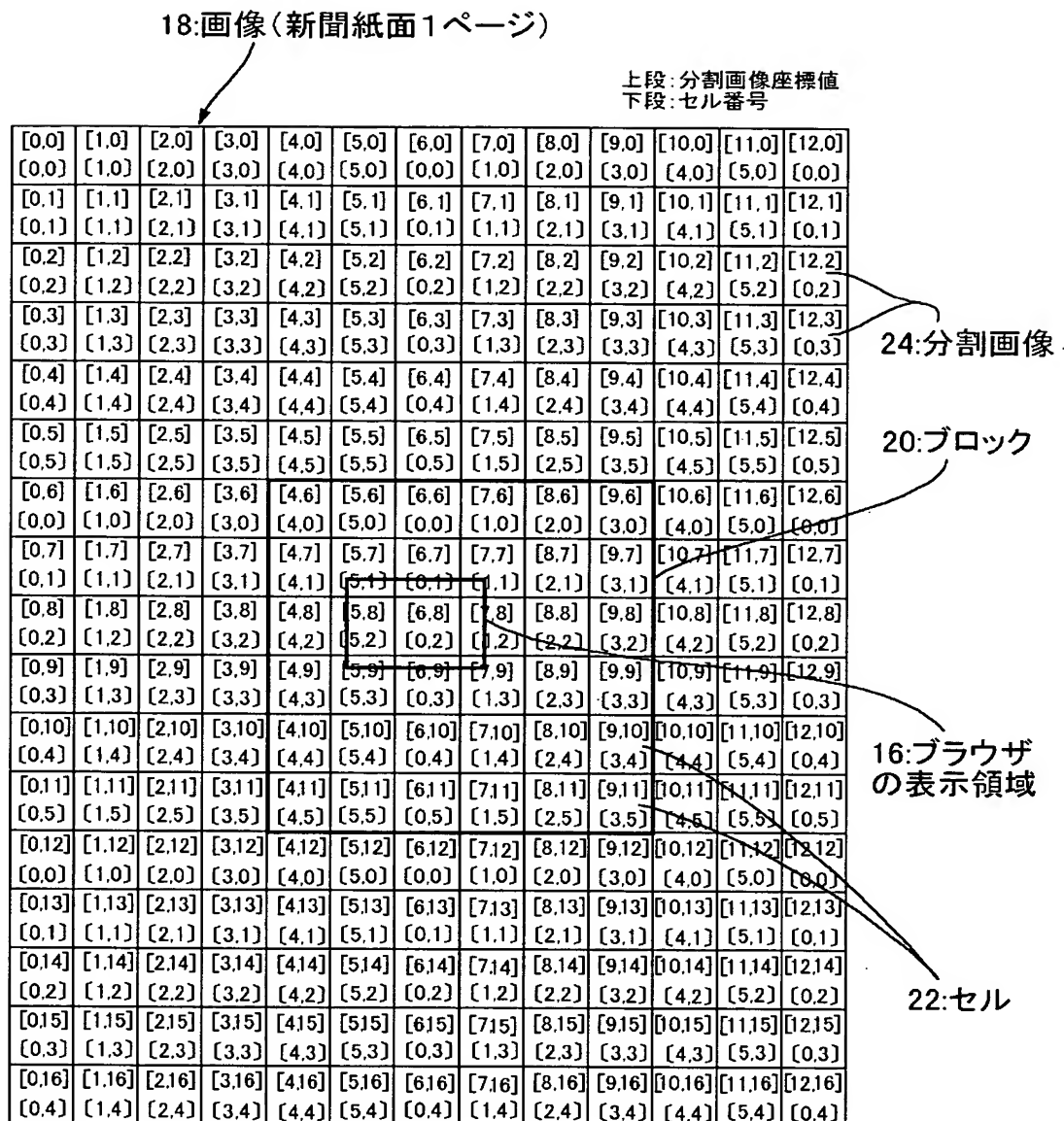
【図 8】

22:セル

20:ブロック

(-563, -907)	(-83, -907)	(397, -907)	(877, -907)	(1357, -907)	(1837, -907)
(-563, -427)	(-83, -427)	(397, -427)	(877, -427)	(1357, -427)	(1837, -427)
(-563, 53)	(-83, 53)	(397, 53)	(877, 53)	(1357, 53)	(1837, 53)
(-563, 533)	(-83, 533)	(397, 533)	(877, 533)	(1357, 533)	(1837, 533)
(-563, 1013)	(-83, 1013)	(397, 1013)	(877, 1013)	(1357, 1013)	(1837, 1013)
(-563, 1493)	(-83, 1493)	(397, 1493)	(877, 1493)	(1357, 1493)	(1837, 1493)

【図 9】



【図 1 0】

```

function setImageSell(){
    originX =  $X_{b0}$ ;      // ブロックの最初セルのX軸原点(式19)
    originY =  $Y_{b0}$ ;      // ブロックの最初セルのY軸原点(式20)
    for(x=0; x<= $S_{xl}$ ; x++){ // 画像のX軸方向の最初の分割画像から最後の分割画像まで

        for(y=0; y<= $S_{yl}$ ; y++){ // 画像のY軸方向の最初の分割画像から最後の分割画像まで

            // ブロックのX軸最初セルから最後セル間の時実行

            if(x >=  $S_{xl}$  && x <=  $S_{xl}$ ){

                sellX = x % 6;      // xを6で割った余りをX軸方向のセル番号として使用する。
                // ブロックのY軸最初セルから最後セル間の時実行

                if(y >=  $S_{yl}$  && y <=  $S_{yl}$ ){

                    sellY = y % 6;      // yを6で割った余りをY軸方向のセル番号として使用する。
                    // 各セル (canvasXY) に画像データ (imageXY) を格納する。
                    // 実際は CGI によってサーバ側に画像データを要求することで実行される。
                    document.all("canvas" + sellX + sellY).innerHTML = "<IMG ID= 'image" +
                        x + y + "' SRC='image.cgi?a=xxx&b=xxx&c=xxx&d=xxx&e=xxx&f=
                        xxx&g=xxx' STYLE='left:" + originX + ";top:" + originY + ";'>";

                }

            }

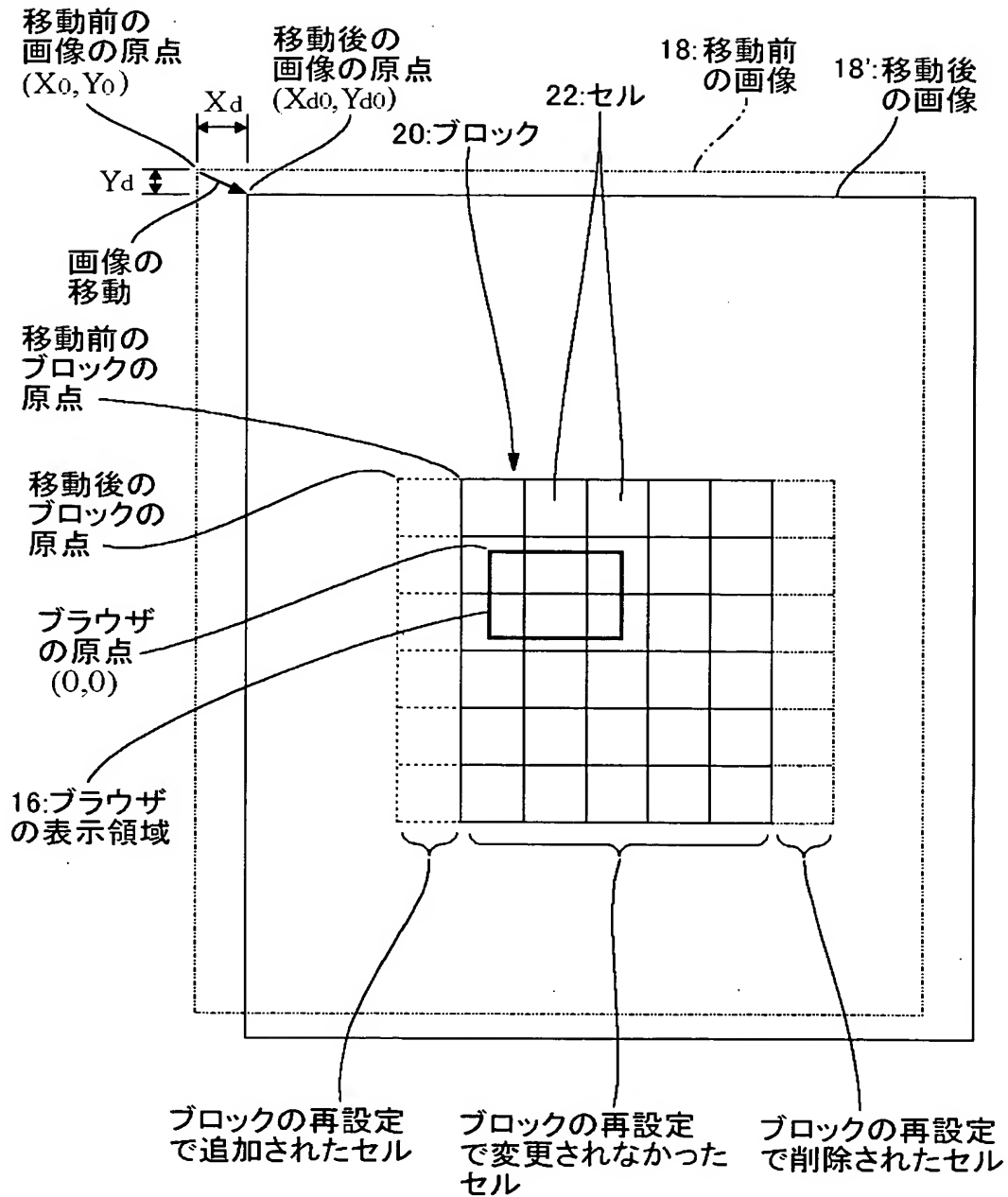
            originY += 480; // 4 8 0 を足すことで、次のセルの Y 軸原点を設定する。
        }

        originX += 480; // 4 8 0 を足すことで、次のセルの X 軸原点を設定する。
        originY =  $Y_{b0}$ ; // Y軸原点をY軸最初セルのY軸原点に戻す。
    }

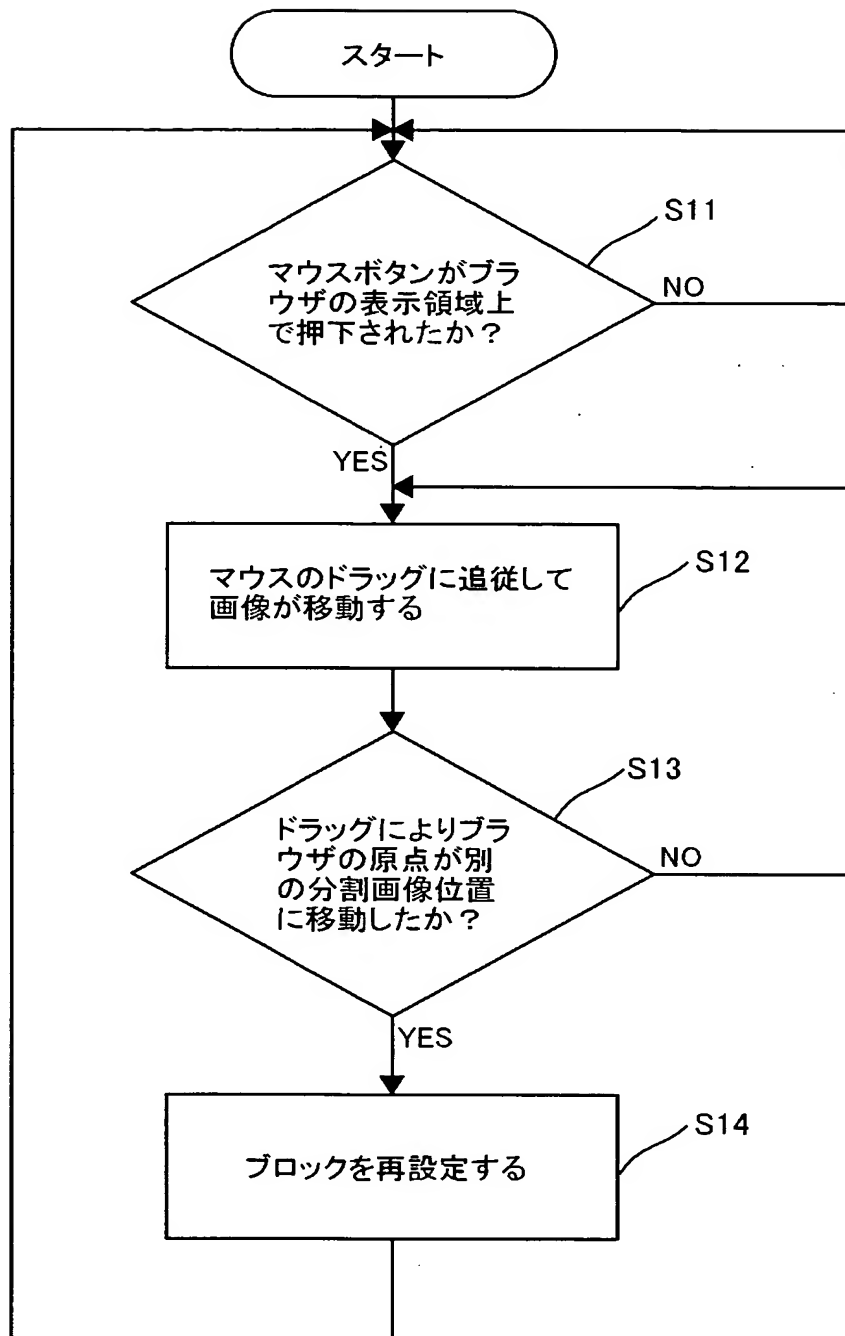
    originX =  $X_{b0}$ ; // X軸原点をX軸最初セルのX軸原点に戻す。
}

```

【図 11】



【図 12】



## 【図 1 3】

```
document.onmousedown = setMouseDown; //マウスがダウンされたとき
document.onmousemove = setMouseMove; //マウスが移動したとき

function setMouseDown(){
    downX    = event.x; //クリックされたX軸座標
    downY    = event.y; //クリックされたY軸座標
    originX  = downX - document.all.canvasBlock.style.pixelLeft;
    originY  = downY - document.all.canvasBlock.style.pixelTop;
}

function setMouseMove(){
    dragX    = event.x; // 移動している時のX軸座標
    dragY    = event.y; // 移動している時のY軸座標
    // ブロックの原点を再計算する。
    document.all.canvasBlock.style.pixelLeft = dragX - originX;
    document.all.canvasBlock.style.pixelTop  = dragY - originY;
}
```



【図 14】

上段: 分割画像座標値  
下段: セル番号

[4.6]	[5.6]	[6.6]	[7.6]	[8.6]	[9.6]
[4.0]	[5.0]	[0.0]	[1.0]	[2.0]	[3.0]
[4.7]	[5.7]	[6.7]	[7.7]	[8.7]	[9.7]
[4.1]	[5.1]	[0.1]	[1.1]	[2.1]	[3.1]
[4.8]	[5.8]	[6.8]	[7.8]	[8.8]	[9.8]
[4.2]	[5.2]	[0.2]	[1.2]	[2.2]	[3.2]
[4.9]	[5.9]	[6.9]	[7.9]	[8.9]	[9.9]
[4.3]	[5.3]	[0.3]	[1.3]	[2.3]	[3.3]
[4.10]	[5.10]	[6.10]	[7.10]	[8.10]	[9.10]
[4.4]	[5.4]	[0.4]	[1.4]	[2.4]	[3.4]
[4.11]	[5.11]	[6.11]	[7.11]	[8.11]	[9.11]
[4.5]	[5.5]	[0.5]	[1.5]	[2.5]	[3.5]

20: ブロック

22: セル

(a) 画像を移動する前

上段: 分割画像座標値  
下段: セル番号

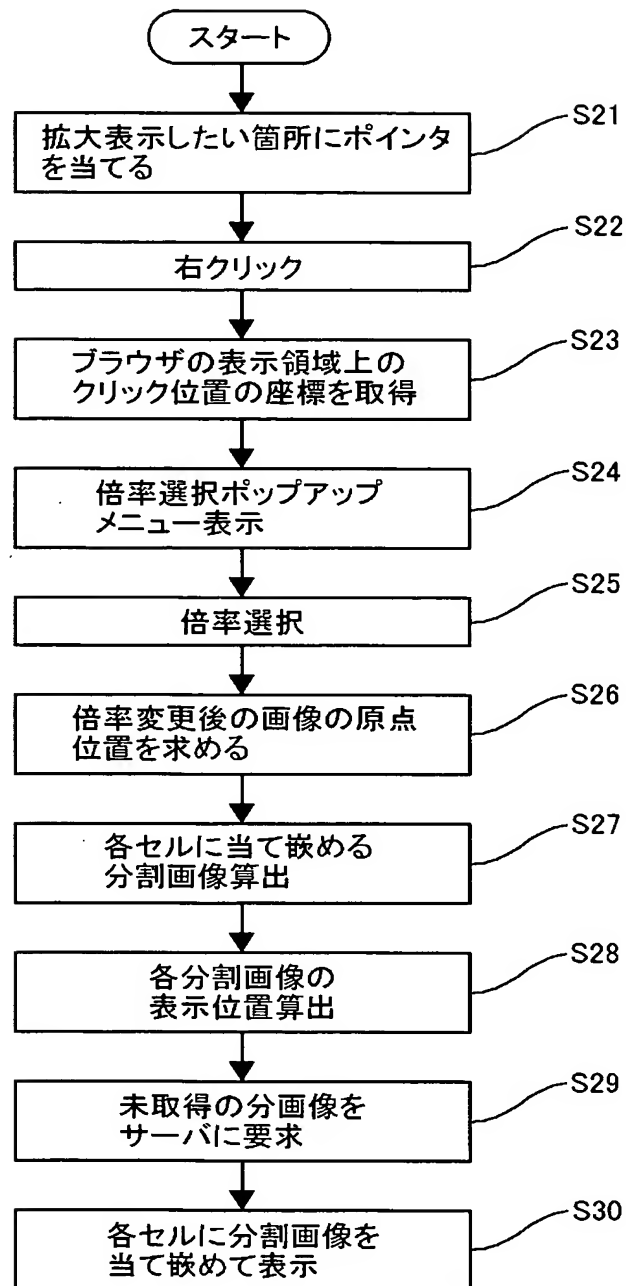
[3.6]	[4.6]	[5.6]	[6.6]	[7.6]	[8.6]
[3.0]	[4.0]	[5.0]	[0.0]	[1.0]	[2.0]
[3.7]	[4.7]	[5.7]	[6.7]	[7.7]	[8.7]
[3.1]	[4.1]	[5.1]	[0.1]	[1.1]	[2.1]
[3.8]	[4.8]	[5.8]	[6.8]	[7.8]	[8.8]
[3.2]	[4.2]	[5.2]	[0.2]	[1.2]	[2.2]
[3.9]	[4.9]	[5.9]	[6.9]	[7.9]	[8.9]
[3.3]	[4.3]	[5.3]	[0.3]	[1.3]	[2.3]
[3.10]	[4.10]	[5.10]	[6.10]	[7.10]	[8.10]
[3.4]	[4.4]	[5.4]	[0.4]	[1.4]	[2.4]
[3.11]	[4.11]	[5.11]	[6.11]	[7.11]	[8.11]
[3.5]	[4.5]	[5.5]	[0.5]	[1.5]	[2.5]

20: ブロック

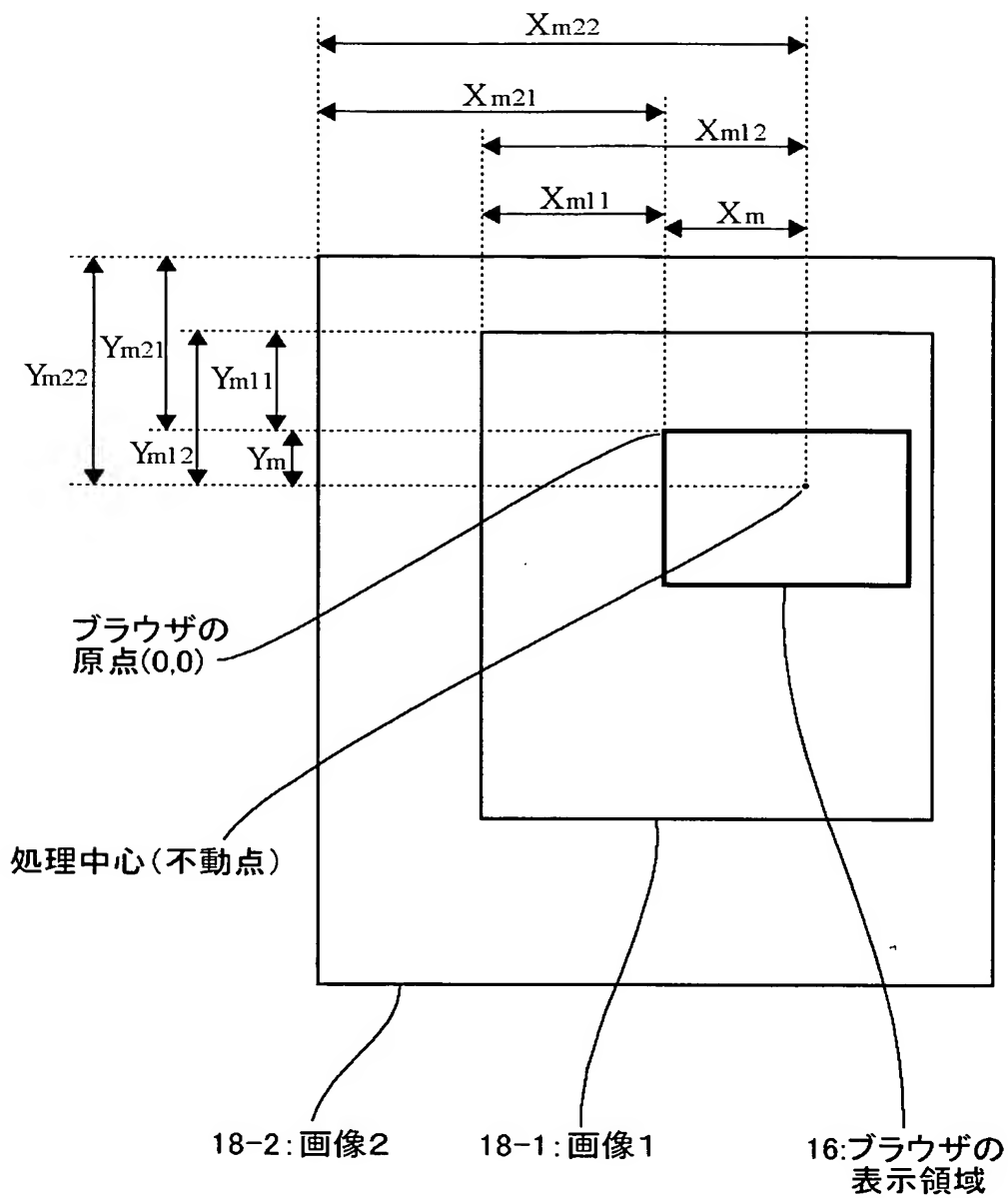
22: セル

(b) 画像を移動した後

【図 15】



【図 16】



【書類名】 要約書

【要約】

【課題】 横縦のサイズが大きい画像をダウンロードして、少ない待ち時間でビューアに表示できるようにする。

【解決手段】 ビューア 12 の表示領域よりも大きい画像を縦横複数の領域に分割してサーバ 10 に予め用意する。ビューア 12 は、該画像とビューア 12 の表示領域との相対位置に応じて、ビューア 12 の表示領域に少なくとも一部が入る複数の分割画像を判断して、該分割画像をサーバ 10 に優先的に要求する。サーバ 10 は該要求に応じた複数の分割画像を優先的に送信する。ビューア 12 は受け取った分割画像を元の並び状態に配列して表示する。

【選択図】 図 1

特願 2 0 0 3 - 1 1 2 9 9 8

出 願 人 履 歴 情 報

識別番号

[ 5 9 4 1 6 4 3 7 9 ]

1. 変更年月日

1 9 9 4 年 9 月 8 日

[変更理由]

新規登録

住 所

東京都豊島区南大塚 3 - 2 0 - 6

氏 名

株式会社サピエンス